UNSUPERVISED FEATURE ANALYSIS FOR HIGH DIMENSIONAL BIG DATA

BY

MINGJIE QIAN

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2015

Urbana, Illinois

Doctoral Committee:

Professor Chengxiang Zhai, Chair
Professor Jiawei Han
Professor Dan Roth
Doctor Liangjie Hong, Yahoo Labs

Abstract

In practice we often encounter the scenario that label information is unavailable due to either high cost of manual labeling or unwillingness of users to label. When label information is not available, traditional supervised learning can not be directly applied so we need to study unsupervised methods which could work well even without supervision.

Feature analysis has been proven effective and important for many applications. Feature analysis is a broad research field, whose research topics includes but are not limited to feature selection, feature extraction, feature construction, and feature composition e.g., in topic discovery the learned topics can be viewed as compound features. In many real systems, it is often necessary and important to do feature analysis to determine which individual or compound features should be used for posterior learning tasks. The effectiveness of traditional feature analysis often relies on labels of the training data examples. However, in the era of big data, label information is often unavailable. In the unsupervised scenario, it is more challenging to do feature analysis.

Two important research topics in unsupervised feature analysis are unsupervised feature selection and unsupervised feature composition, e.g., to discover topics as compound features. This would naturally create two lines for unsupervised feature analysis. Also, combined with single-view or multiple-view for the data, we would generate a table with four cells. Except for the single-view feature composition (or topic discovery) where there're already many work done e.g., PLSA, LDA, and NMF, the other three cells correspond to new research topics, and there is few work done yet.

For single view unsupervised feature analysis, we propose two unsupervised feature selection methods. For multi-view unsupervised feature analysis, we focus on text-image web news data and propose a multi-view unsupervised feature selection method and a text-image topic model.

Specifically, for single-view unsupervised feature selection, we propose a new method that is

called Robust Unsupervised Feature Selection (RUFS), where pseudo cluster labels are learned via local learning regularized robust NMF and feature selection is performed simultaneously by robust joint $l_{2,1}$-norm minimization. Outliers could be effectively handled and redundant or noisy features could be effectively reduced. We also design a (projected) limited-memory BFGS based linear time iterative algorithm to efficiently solve the optimization problem.

We also study how the choice of norms for data fitting and feature selection terms affect the ultimate unsupervised feature selection performance. Specifically, we propose to use joint adaptive loss and l_2/l_0 minimization for data fitting and feature selection. We mathematically explain desirable properties of joint adaptive loss and l_2/l_0 minimization over recent unsupervised feature selection models. We solve the optimization problem with an efficient iterative algorithm whose computational complexity and memory cost are linear to both sample size and feature size.

For multiple-view unsupervised feature selection, we propose a more effective approach for high dimensional text-image web news data. We propose to use raw text features in label learning to avoid information loss. We propose a new multi-view unsupervised feature selection method in which image local learning regularized orthogonal nonnegative matrix factorization is used to learn pseudo labels and simultaneously robust joint $l_{2,1}$-norm minimization is performed to select discriminative features. Cross-view consensus on pseudo labels can be obtained as much as possible.

For multi-view topic discovery, we study how to systematically mine topics from high dimensional text-image web news data. The application problem is important because almost all news articles have one picture associated. Unlike traditional topic modeling which considers text alone, the new task aims to discover heterogeneous topics from web news of multiple data types. We propose to tackle the problem by a regularized nonnegative constrained $l_{2,1}$-norm minimization framework. We also present a new iterative algorithm to solve the optimization problem.

The proposed single-view feature selection methods can be applied on almost all single-view data. The proposed multi-view methods are designed to process text-image web news data, but the idea can be naturally generalized to analyze any multi-view data. Practitioners could run the proposed methods to select features that will be used in posterior learning tasks. One can also run our multi-view topic model to analyze and visualize topics in text-image web news corpora to help interpret the data.

To my parents.

Acknowledgments

First of all, I would like to express my sincere gratitude to my advisor Prof. Chengxiang Zhai for his continuous support to my Ph.D. study and his great supervision and guidance over the past five years. His enthusiasm for research and pursuit of dream motivate me to conquer the fear when I walk through the darkness. His broad knowledge and profound insight always inspire me in all aspects of research.

I appreciate my thesis committee members Prof. Jiawei Han, Prof. Dan Roth, and Dr. Liangjie Hong for their insightful comments and constructive suggestions. Dr. Liangjie Jie Hong was my mentor when I did my summer internship at Yahoo Labs. He led me and triggered my research interests to personalization and recommendation.

I'm also grateful to my friends Prof. Feiping Nie, Prof. Yangqiu Song, and Prof. Quanquan Gu for their invaluable suggestions and inspiring comments on my research papers.

I also want to thank Prof. Shanfeng Zhu in Fudan University for our discussions on multiple research topics on bioinformatics during his visit at UIUC.

I would like to thank Dr. Suju Rajan and Dr. Junling Hu for their supervision for my summer internships at Yahoo Labs and eBay respectively. Their vision and thoughts influence me and help shape my future career in industry.

Many thanks to my fellow labmates in TIMan group including Yue Lu, Yuanhua Lv, Duo Zhang, Hyun Duk Kim, Hongning Wang, Dae Hoon Park, Yanen Li, Huizhong Duan, Xiaolong Wang, Yinan Zhang, Xueqing Liu, Rongda Zhu, Sheng Wang, Shan Jiang, Sean Massung, Chase Geigle, Ismini Lourentzou and many others.

Most importantly, I will always thank my parents Qiushi Li and Zhigang Qian for their encouragement, without whom I would not be able to overcome the difficulties.

Table of Contents

Chapter 1

Introduction

Feature analysis has been proven effective and important for many applications in industry. Feature analysis is a broad research field, whose research topics includes but are not limited to feature selection, feature extraction, feature construction, and feature composition e.g., in topic discovery the learned topics can be viewed as compound features. In many real systems, it is often necessary and important to do feature analysis to decide which individual or compound features we should use to represent the data points for learning tasks. The effectiveness of traditional feature analysis often relies on labels of the training data examples. However, in the era of big data, label information are often unavailable due to either high cost of manual labeling or unwillingness of users to label. In the unsupervised scenario, it is more challenging to do feature analysis. This thesis attempts to study feature analysis methods that can work without supervision.

In addition to the scarcity of label information, big data usually faces two other challenges. The first is the curse of dimensionality. There are several undesirable consequences of curse of dimensionality. E.g., the number of the data examples required for training increases exponentially, nearly all of the high-dimensional space is far away from the center, and the majority of the data space is unseen to the target function which reduces the predictive power of a learning model, leading to over-fitting. It is therefore important and necessary to reduce dimensionality. This can be usually achieved by feature selection and feature learning. Feature selection tries to select the most useful features to represent the original data points whereas feature learning aims to learn mapping functions from original features to new representations. Compared to feature learning, one advantage of feature selection is the high interpretability of results since it preserves the physical sense of the original features.

The second challenge is heterogeneity (or multiple view/modality). Unlabeled big data are

Table 1.1: Overall contributions in this thesis.

	Unsupervised Feature Selection	Unsupervised Feature Composition discover topics as "compound features"
Single-View	IJCAI 13, IJCNN 15	PLSA [1], LDA [2] and NMF [3] etc.
Multiple-View	CIKM 14	ECIR 14

usually composed of multiple views. For example, it is prevalent that web news articles contain both text content and images. The heterogeneity of big data is challenging because different data types have different properties, and pseudo labels learned from different views often conflict with each other. For big data when label information is out of reach, unsupervised feature analysis approaches are needed. For example, unsupervised feature selection for multi-view data is desirable for unsupervised feature selection on data of multiple views, or topic discovery methods for multi-view data are needed to effectively and efficiently mine the latent structures from multi-view data, which would be very helpful to automatically organize the data, analyze the knowledge, and explore the data hidden structure.

This thesis attempts to study two important research topics in unsupervised feature analysis: unsupervised feature selection and unsupervised feature composition, e.g., to discover topics as compound features. This would naturally create two lines for unsupervised feature analysis. Also, combined with single-view or multiple-view for the data, we would generate a table with four cells shown in Table 1.1. Except for the single-view feature composition (or topic discovery) where there're already many work done e.g., PLSA, LDA, and NMF, the other three cells correspond to new research topics, and there is few work done yet. This thesis attempts to deal with the three new cells in the table by developing novel unsupervised approaches for feature selection, topic discovery, and exploring their applications. We show how feature selection can be done in a unsupervised scenario. We propose unsupervised feature selection methods that don't rely on labels and could be applied on single view and multiple view data. We also propose a new unsupervised method that could mine the latent structure from multi-view data. Although we use only text-image web news articles for evaluation, the formulation can be naturally extended for general multi-view data.

For single-view unsupervised feature selection, we first propose a new unsupervised feature selection method, i.e., Robust Unsupervised Feature Selection (RUFS). Unlike traditional unsu-

pervised feature selection methods such as MCFS [4], UDFS [5] and NDFS [6], pseudo cluster labels are learned via local learning regularized robust nonnegative matrix factorization. During the label learning process, feature selection is performed simultaneously by robust joint $l_{2,1}$ norms minimization. Since RUFS utilizes $l_{2,1}$ norm minimization on processes of both label learning and feature learning, outliers and noise could be effectively handled and redundant or noisy features could be effectively reduced. Our method adopts the advantages of robust nonnegative matrix factorization, local learning, and robust feature learning. In order to make RUFS scalable, we design a (projected) limited-memory BFGS based iterative algorithm to efficiently solve the optimization problem of RUFS in terms of both memory consumption and computation complexity. Experimental results on six benchmark real world data sets show that RUFS outperforms the state-of-the-art methods in terms of both clustering and classification settings.

We also want to study how the choice of norms for data fitting and feature selection terms affect the ultimate unsupervised feature selection performance. Specifically, we propose to use joint adaptive loss and l_2/l_0 minimization for data fitting and feature selection. We mathematically explain desirable properties of joint adaptive loss and l_2/l_0 minimization over recent unsupervised feature selection models. We solve the optimization problem with an efficient iterative algorithm and prove that all the expected properties of unsupervised feature selection can be preserved. We also show that the computational complexity and memory use is only linear to the number of instances and square to the number of clusters. Experiments show that our algorithm outperforms the state-of-the-arts on seven real world benchmark data sets.

For multiple-view unsupervised feature selection, we propose a more effective unsupervised feature selection approach for high dimensional multi-view data. Although we use text-image web news data for evaluation due to the prevalence of this kind of data nowadays, the idea of the proposed method can be naturally generalized to analyze any multi-view data. The fact that unlabeled high-dimensional text-image web news data are produced every day presents new challenges to unsupervised feature selection on multi-view data. State-of-the-art multi-view unsupervised feature selection methods such as AUMFS [7] and MVFS [8] learn pseudo class labels by spectral analysis, which is sensitive to the choice of similarity metric for each view. For text-image data, the raw text itself contains more discriminative information than similarity graph which loses information during

construction, and thus the text feature can be directly used for label learning, avoiding information loss as in spectral analysis. We propose a new multi-view unsupervised feature selection method in which image local learning regularized orthogonal nonnegative matrix factorization is used to learn pseudo labels and simultaneously robust joint $l_{2,1}$-norm minimization is performed to select discriminative features. Cross-view consensus on pseudo labels can be obtained as much as possible. We systematically evaluate the proposed method in multi-view text-image web news datasets. Our extensive experiments on web news datasets crawled from two major US media channels: CNN and FOXNews demonstrate the effectiveness of the new method over state-of-the-art multi-view and single-view unsupervised feature selection methods.

Single-view feature composition (topic discovery) has been well studied in the last decade, so we focus on multi-view topic discovery. We study how to systematically mine topics from high dimensional text-image web news data. We formally propose a new application problem: unsupervised text-image topic discovery. The application problem is important because almost all news articles have one picture associated. Unlike traditional topic models like PLSA [1] and LDA [2] which consider text alone, the new task aims to discover heterogeneous topics from web news of multiple data types. The heterogeneous topic discovery is challenging because different media data types have different characteristics and structures, and a systematic solution that can integrate information propagation and mutual enhancement between data of different types in a principled way is not trivial, especially when no supervision information is available. We propose to tackle the problem by a regularized nonnegative constrained $l_{2,1}$-norm minimization framework. We also present a new iterative algorithm to solve the optimization problem. To objectively evaluate the proposed method, we collect two real world text-image web news datasets. Experimental results show the effectiveness of the new approach.

The proposed single-view feature selection methods can be applied on almost all single-view data. The proposed multi-view methods are designed to process text-image web news data, but the idea can be naturally generalized to analyze any multi-view data. Practitioners could run the proposed methods to select features that will be used in posterior learning tasks. E.g., we can do unsupervised feature selection in the first phase to select a subset of features which is used to represent the data points as input of posterior learning systems. One can also apply our multi-view

topic model to analyze and visualize topics in text-image web news corpora.

Chapter 2

Related Work

Feature selection has attracted increasing attention in recent years, and many feature selection algorithms have been proposed, which can be grouped into three families: filter, wrapper, and embedded methods. Filter methods [9][10][11][12][13][14][5] select a subset of features by leveraging statistical properties of data, and are usually performed before applying classification algorithms. For wrapper methods [15][16][17], feature selection is wrapped in a learning algorithm and the classification performance on selected features is taken as the evaluation criterion. Embedded approaches [18][19][20] perform feature selection when training the models. Wrapper and embedded methods couples feature selection with built-in classifiers tightly, which lead to less generality and extensive computation. We thus adopt the filter approach in this thesis.

From the perspective of label availability, feature selection algorithms can also be classified into supervised feature selection and unsupervised feature selection. Supervised feature selection methods, such as [10][21][22][23], are usually able to effectively select good features since labels of training data, which contain the essential discriminative information for classification, can be used. However, in unsupervised scenario, label information is unavailable directly, which makes the task of feature selection more challenging.

Several unsupervised feature selection algorithms are proposed recently. A commonly used criterion in unsupervised feature learning is to select features best preserving data similarity or manifold structure constructed from the whole feature space [11][12][4], but they fail to incorporate discriminative information implied within data, though it has been shown to be important in data analysis [24]. Earlier unsupervised feature selection algorithms evaluate the importance of each feature individually and select feature one by one [11][12], with a limitation that correlation among features is neglected pointed by [22][4] which applied two-step approaches, i.e., spectral regression

to unsupervised feature selection. [25] is also related to unsupervised feature selection. It proposes a row-wise sparse subspace learning method to improve subspace learning performance. Modern unsupervised feature selection algorithms perform feature selection by simultaneously exploiting discriminative information and feature correlation. Unsupervised Discriminative Feature Selection (UDFS) [5] aims to select the most discriminative features for data representation, where manifold structure is also considered. However, its orthogonal constraint on the feature selection projection matrix is unreasonable since feature weight vectors are not necessarily orthogonal with each other in nature. Nonnegative Discriminative Feature Selection (NDFS) [6] performs nonnegative spectral analysis and feature selection simultaneously. One factor that is ignored in both UDFS and NDFS is that data is usually not ideally clean, and outliers or noise often exist in it. UDFS and NDFS are not robust and are vulnerable to outliers or noise. Another deficiency of UDFS and NDFS is that their computation complexity is cubic to the number of features which severely limits their applicability on high dimensional data, e.g., text data and genetic data.

Since the most discriminative information for feature selection is usually encoded in labels, it is very important to predict a good cluster indicators as pseudo labels for unsupervised feature selection [6]. Another important factor which effects the performance of feature selection is the consideration of outliers and noise [21]. Real data is usually not ideally distributed, outliers and noise often appear in the data, thus it is important or even necessary to consider robustness for unsupervised feature selection.

There are also some unsupervised feature selection method proposed on multi-view data recently. State-of-the-art unsupervised feature selection methods [7, 8] for multi-view data use spectral clustering across different views to learn the most consistent pseudo class labels and simultaneously use the learned labels to do feature selection. More specifically, Adaptive Unsupervised Multi-view Feature Selection (AUMFS) [7] uses spectral clustering on a combined data similarity graph from different views to learn the labels that have most consensus across different views, and then use $l_{2,1}$-norm regularized robust sparse regression to learn one weight matrix for all the features of different views to best approximate the cluster labels. [8] presents a new unsupervised multi-view feature selection method called Multi-View Feature Selection (MVFS). MVFS also uses spectral clustering on the combined data similarity graph from different views to learn the labels, but learn one weight

matrix for each view to best fit the learned pseudo class labels by joint squared Frobenius norm (fitting term) and $l_{2,1}$-norm (rowise sparsity-inducing). Both [7] and [8] share the disadvantage that they're sensitive to the combined data similarity graph, especially when there are quite a number of unrelated and noisy features in the feature space, and there is information loss during graph construction.

Now we describe related work on topic discovery from both single-view and multi-view perspectives. Topic discovery/modeling/mining is a popular research area in information retrieval, data mining and knowledge management, which discovers meaningful latent patterns (e.g., metadata, topics, and events that are instances of topics). Topic mining is very useful and has a promising perspective for web applications, especially in an era of fast growth of web information such as multimedia news, web blogs, social network and twitters.

Although a number of topic mining methods have been proposed, including both generative models [26, 2] and discriminative models [3, 27], they mainly focus on discovering topics from a single type of data, i.e., text data. There are two groups of approaches for text based topic mining.

The first group tries to build generative probabilistic topic models to learn the latent topic concepts. The representative works in this line are Probabilistic Latent Semantic Analysis (PLSA) [26] and Latent Dirichlet Allocation (LDA) [2].

PLSA models each word in a document as a sample from a mixture model, where the mixture components are multinomial random variables that represent topics. Each document is represented as a list of mixing weights for these mixture components and thereby reduced to a probability distribution on a fixed set of topics. This distribution is the reduced representation associated with the document. PLSA suffers from over-fitting problems since its number of parameters grows linearly with the number of documents.

To address the over-fitting problems of PLSA, LDA was proposed, which assumes that the probability distributions of documents over topics are generated from the same Dirichlet distribution. In this sense, LDA can be seen as Bayesian PLSA. The basic idea is that documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words. Many further extensions of PLSA and LDA have been proposed (see, e.g., [28, 29]).

The second group aims to decompose the data matrix algebraically subject to some criteria.

This line of work assumes that the latent topic information lies in the input data matrix. By approximating the data matrix by a multiplication of several matrices components under some constraints, a topic model matrix can be learned. Thus, documents can be approximately represented by a linear combination of topic vectors. The advantage of matrix-factorization based topic mining methods is the high flexibility and avoidance of independent assumptions.

Nonnegative matrix factorization (NMF) [3] is an important branch in this family, which tries to find a good matrix factorization using a multiplication of two nonnegative matrices.

In multimedia field there have been some studies working on combined text and visualized features for web document retrieval, image clustering, image classification and image retrieval. In [30], latent semantic indexing (LSI) [31] is used, together with textual and visual features, for content-based web document retrieval. LSI seeks to find the latent correlation between terms and documents and obtains a good approximation of document features in order to get a better retrieval performance. However, it cannot discover interpretable topics (positively weighted term vectors). The number of singular values is chosen to be the one getting the best retrieval results, not necessarily the number of topics (In [30], 12 highest singular values were chosen whereas the number of topics is 4). Besides, the negative components for the latent semantic vectors are not reasonable for topic representation. Some image clustering works [32, 33] also utilize text features to help clustering image collections, however, they aim to optimize the image clustering performance rather than explicitly learn a joint text-image topic representation. Reference [34] exploits the link structure of the web graph to learn an image-classification model for multimedia data. However, their work does not attempt to discover subtopics buried in text data, and their target is different from our task in that the semantic topic is implicitly learned in the image-classifier whereas we focus on explicitly learning a good topic representor. Recently, text information is used for several image retrieval works [35, 36], however, their task is different from ours, and their work cannot discover joint text-image topics.

There have also been some works on automatic image annotation [37, 38]. In these works, image regions are taken as visual words, and a generative model is then constructed under some assumptions of independence for generating the pair of an image and its caption. Their task is also different from ours. The automatic image annotation aims to find a generative model to fit

the image region and its annotation, our task is to discover coherent topics within a text-image collection, and our framework can incorporate any type of data though in this work we focus on text-image domain for simplicity.

Multi-view learning, which aims to learn better models to cluster data in multiple views, is a machine learning research area that can be applied for multi-view topic discovery, but state-of-the-art multi-view learning methods cannot do this task very well. Co-trained multi-view spectral clustering [39] iteratively uses the spectral embedding from one view to constrain the similarity graph used for the other view. However, this approach heavily relies on similarity graph for each view and completely ignores the detailed information, which may badly hurt the clustering performance due to loss of discriminative information. Also it's not straight forward to generate multi-view topic representation via this approach. [40] proposes to generalize K-means for multi-view data clustering. However, its performance tends to be dominated by the worst domain since the algorithm will assign large weight to the domain with the largest approximation error as will be demonstrated in the experiment later. There're also some heterogeneous data co-clustering work [41][42], however they require some supervision information. For example, [41] require user specified must-link and cannot-link constraint in the central type, and [42] require user preference before clustering. Besides, although heterogeneous co-clustering methods appear to be able to tackle our problem, they do not aim to explicitly learn representative and interpretable multi-view topics from heterogenous web news data. The major goal of our work is to provide an effective multi-view learning approach to discover text-image topics from web news data without any supervision.

Now we have described the related works for the thesis topic. Because the idea of NMF plays an important role in almost all thesis work, we will elaborate it in the next section so that it would be easier for readers to understand the objective functions throughout this thesis.

2.1 Nonnegative Matrix Factorization Family

NMF and its regularized variations are widely used methods in the field of machine learning [3], data mining [27] [43] [44] [45] [46], and information retrieval [47] [48]. This group of methods try to find a good matrix factorization using a multiplication of two nonnegative matrices. It can be shown

to be equivalent to spectral clustering [49]. In the area of topic mining, NMF is closely related to PLSA [26]. It has been shown in [50] [51] that if KL divergence cost function is used other than squared errors, the objective function of PLSA is identical to that of NMF except for a constant, which means that NMF multiplicative update algorithm and the EM algorithm for training PLSA are alternative approaches to optimize the equivalent objective functions. For important NMF extensions proposed in literature, please refer to [27, 44, 45, 46, 52, 41, 53].

The general form of regularized NMF can be formulated as below:

$$\min_{\mathbf{F}\geq 0,\mathbf{G}\geq 0} \left\|\mathbf{X}-\mathbf{F}\mathbf{G}^T\right\|_F^2 + \mathcal{F}\left(\mathbf{G}\right) + \mathcal{G}\left(\mathbf{F}\right). \tag{2.1}$$

Here $\mathcal{F}$ and $\mathcal{G}$ can be any linear or quadratic convex functions with finite lower bound (e.g., l_1-norm, Frobenius norm, and graph Laplacian regularization) , and they are general enough to cover existing and potential regularization terms. We use alternate updating technique to get a local optimum. Given fixed nonnegative $\mathbf{F}$, the optimization problem of $\mathbf{G}$ is

$$\min_{\mathbf{G}\geq 0} \left\|\mathbf{X}-\mathbf{F}\mathbf{G}^T\right\|_F^2 + \mathcal{F}\left(\mathbf{G}\right). \tag{2.2}$$

Likewise, given fixed $\mathbf{G}$, the optimization problem of F is

$$\min_{\mathbf{F}\geq 0,} \left\|\mathbf{X}^T-\mathbf{G}\mathbf{F}^T\right\|_F^2 + \mathcal{G}\left(\mathbf{F}\right). \tag{2.3}$$

We see from Eq.(2.2) and Eq.(2.3) that once $\mathbf{G}$ is solved $\mathbf{F}$ can be solved following the same rule by replacement of counterpart matrices. In literature, there are several algorithms to solve NMF. The original work use multiplicative update methods [3]. There are other multiplicative update methods proposed in [27, 44]. Recently, several methods are proposed based on the Alternating Nonnegative Least Squares(ANLS) framework suggested by [54]. This framework has been proved to converge to stationary points by [55] so long as each nonnegative least square sub-problem can be solved exactly. Among these methods are projected gradient method [56], active set method [57], projected Newton method [58], and modified active set method called Block-Pivot [59]. Now, the state-of-the-art technique is to use coordinate descent methods, which update one variable at a

time. Coordinate descent methods have been tested efficient for NMF in [60, 61] and can be made successful in large scale problems.

2.1.1 Robust Nonnegative Matrix Factorization Using $l_{2,1}$-norm

One of the most important drawbacks of standard NMF is that it is prone to outliers since standard NMF uses squared residue error for each data point. Thus a few outliers with large errors may dominate the objective function due to the squared residue errors, which may further lead to unexpected effect on the base matrix and label indicator matrix. Thus it is very necessary to present robust NMF formulation and discuss its properties. One strategy to consider outliers in NMF is using the $1_{2,1}$-norm [62], which can be formulated as

$$\min \|\mathbf{X} - \mathbf{FG}\|_{2,1} \quad \text{s.t.} \quad \mathbf{F} \geq 0,\ \mathbf{G} \geq 0. \tag{2.4}$$

Note that here $l_{2,1}$-norm is used in the objective function so that only absolute residue errors are to be minimized thus it's harder for outliers to dominate the objective function. Experiments on a bunch of benchmark datasets show that the robust NMF provides more faithful basis factors and consistently better clustering results as compared to standard NMF [62].

For optimization, similar to [27, 52] auxiliary functions are proposed to derive multiplicative updating rules for the basis matrix and label indicator matrix in the original paper. However, the optimization strategy is limited in that the optimization approach cannot be directly applied to cases where regularization terms are involved. We will propose later in the thesis a more general optimization strategy that is able to solve both robust NMF and its regularized variants.

We will extensively apply $l_{2,1}$-norm in this thesis. In the first chapter, we use it to learn a robust label indicator matrix in label learning and robust feature selection. While in the second chapter, we use it to learn robust topic vectors and robust topic assignment matrix.

Chapter 3

Background

Throughout this thesis, matrices are written as boldface capital letters and vectors are denoted as boldface lowercase letters. For a matrix $\mathbf{M} \in \mathcal{R}^{r\times p}$, its i-th row, j-th column are denoted by $\mathbf{m}^i, \mathbf{m}_j$ respectively. $\|\mathbf{M}\|_F \triangleq \sqrt{\sum_{i=1}^{r}\sum_{j=1}^{p} m_{ij}^2}$ is the Frobenius norm of $\mathbf{M}$ and $\mathrm{Tr}\,[\mathbf{M}] \triangleq \sum_{i=1}^{r} m_{ii}$ is the trace of $\mathbf{M}$ if $\mathbf{M}$ is square. For any matrix $\mathbf{M} \in \mathcal{R}^{r\times p}$, its $l_{2,1}$-norm is defined as

$$\|\mathbf{M}\|_{2,1} = \sum_{i=1}^{r} \sqrt{\sum_{j=1}^{p} m_{ij}^2} = \sum_{i=1}^{r} \left\|\mathbf{m}^i\right\|_2. \tag{3.1}$$

Assume that we have n samples $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n$. Let $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_n]^T$ denote the data matrix with each row being a data feature vector, in which $\mathbf{x}_i \in \mathcal{R}^d$ is the feature descriptor of the i-th sample. Suppose these n data samples are sampled from c classes and denote $\mathbf{Y} = [\mathbf{y}_1, \cdots, \mathbf{y}_n]^T \in \{0,1\}^{n\times c}$, where $\mathbf{y}_n \in \{0,1\}^{c\times 1}$ is the cluster indicator vector for sample $\mathbf{x}_i$. The scaled cluster indicator matrix [5][6] $\mathbf{G}$ is defined as

$$\mathbf{G} = [\mathbf{g}_1, \cdots, \mathbf{g}_n]^T = \mathbf{Y}\left(\mathbf{Y}^T\mathbf{Y}\right)^{-\frac{1}{2}}, \tag{3.2}$$

where $\mathbf{g}_i$ is the scaled cluster indicator of $\mathbf{x}_i$. We thus have

$$\mathbf{G}^T\mathbf{G} = \left(\mathbf{Y}^T\mathbf{Y}\right)^{-\frac{1}{2}}\mathbf{Y}^T\mathbf{Y}\left(\mathbf{Y}^T\mathbf{Y}\right)^{-\frac{1}{2}} = \mathbf{I}_c, \tag{3.3}$$

where $\mathbf{I}_c \in \mathcal{R}^{c\times c}$ is an identity matrix.

For multi-view learning, let $\mathbf{X}_v \in \mathcal{R}^{n\times d_v}$ denote the data matrix in the v-th view where the i-th row $\mathbf{x}_v^i \in \mathcal{R}^{d_v}$ is the feature descriptor of the i-th instance in the v-th view. For text-image web news data, $\mathbf{X}_1$ is text view data matrix, and $\mathbf{X}_2$ is image view data matrix.

3.1 Evaluation

For clustering and unsupervised feature selection, two widely used evaluation metrics to measure clustering performance, i.e., Normalized Mutual Information (NMI) and accuracy (ACC) are used in this thesis.

Given a clustering result, NMI is estimated by

$$NMI = \frac{\sum_{k=1}^{c}\sum_{m=1}^{c} n_{k,m}\log\frac{n\cdot n_{k,m}}{n_k\hat{n}_m}}{\sqrt{\left(\sum_{k=1}^{c} n_k\log\frac{n_k}{n}\right)\left(\sum_{m=1}^{c}\hat{n}_m\log\frac{\hat{n}_m}{n}\right)}}, \tag{3.4}$$

where n_k denotes the number of data contained in the cluster $\mathcal{D}_k$ $(1 \leq k \leq c)$, $\hat{n}_m$ is the number of data belonging to the ground truth class $\mathcal{L}_m$ $(1 \leq m \leq c)$, and $n_{k,m}$ denotes the number of data that are in the intersection between the cluster $\mathcal{D}_k$ and the class $\mathcal{L}_m$. A larger NMI indicates a better clustering result.

Denote q_i as the clustering results and p_i as the ground truth label of $\mathbf{x}_i$. ACC is defined by

$$ACC = \frac{\sum_{i=1}^{n}\delta\left(p_i, map\left(q_i\right)\right)}{n} \tag{3.5}$$

where $\delta(x,y) = 1$ if $x = y; \delta(x,y) = 0$ otherwise, and $map(q_i)$ is the best mapping function that permutes clustering labels to match the ground truth labels using the Kuhn-Munkres algorithm. A larger ACC indicates better performance. Both ACC and NMI range between 0 and 1.

Chapter 4

Robust Unsupervised Feature Selection

In this chapter, we study the problem of feature selection in unsupervised learning from the single-view perspective. We propose a new model by considering the outliers in both labeling learning and feature learning. Additionally, in order to make the proposed method scalable, we design a (projected) limited-memory BFGS based iterative algorithm to efficiently solve the optimization problem in terms of both memory consumption and computation complexity.

4.1 Introduction

The curse of dimensionality is a common phenomenon in many areas, such as pattern recognition, text mining, computer vision [63, 64], and bio-informatics. In practice, not all features are relevant and important to the learning task, many of them are often correlated, redundant, or even noisy sometimes [10][14], which may result in adverse effects such as over-fitting, low efficiency and poor performance. It is therefore important and necessary to reduce dimensionality. This can be usually achieved by transformation to a low dimensional space [65][25] or feature selection. In this chapter, we focus on feature selection, which aims to select discriminative and highly related features and eliminate unrelated, redundant, and noisy features with little or no supervision based on certain criteria.

During recent years, feature selection has attracted increasing attention, and many feature selection algorithms have been proposed, which can be grouped into three families: filter, wrapper, and embedded methods. Filter methods [9][10][11][12][13][14][5] select a subset of features by leveraging statistical properties of data, and are usually performed before applying classification algorithms. For wrapper methods [15][16][17], feature selection is wrapped in a learning algorithm and the classification performance on selected features is taken as the evaluation criterion. Embedded ap-

proaches [18][19][20] perform feature selection when training the models. Wrapper and embedded methods couples feature selection with built-in classifiers tightly, which lead to less generality and extensive computation. We thus adopt the filter approach in this paper.

From the perspective of label availability, feature selection algorithms can also be classified into supervised feature selection and unsupervised feature selection. Supervised feature selection methods, such as [10][21][22][23], are usually able to effectively select good features since labels of training data, which contain the essential discriminative information for classification, can be used. However, in unsupervised scenario, label information is unavailable directly, which makes the task of feature selection more challenging.

Several unsupervised feature selection algorithms are proposed recently. A commonly used criterion in unsupervised feature learning is to select features best preserving data similarity or manifold structure constructed from the whole feature space [11][12][4], but they fail to incorporate discriminative information implied within data, though it has been shown to be important in data analysis [24]. Earlier unsupervised feature selection algorithms evaluate the importance of each feature individually and select feature one by one [11][12], with a limitation that correlation among features is neglected pointed by [22][4] which applied two-step approaches, i.e., spectral regression to unsupervised feature selection. [25] is also related to unsupervised feature selection. It proposes a row-wise sparse subspace learning method to improve subspace learning performance. Modern unsupervised feature selection algorithms perform feature selection by simultaneously exploiting discriminative information and feature correlation. Unsupervised Discriminative Feature Selection (UDFS) [5] aims to select the most discriminative features for data representation, where manifold structure is also considered. However, its orthogonal constraint on the feature selection projection matrix is unreasonable since feature weight vectors are not necessarily orthogonal with each other in nature. Nonnegative Discriminative Feature Selection (NDFS) [6] performs nonnegative spectral analysis and feature selection simultaneously. One factor that is ignored in both UDFS and NDFS is that data is usually not ideally clean, and outliers or noise often exist in it. UDFS and NDFS are not robust and are vulnerable to outliers or noise. Another deficiency of UDFS and NDFS is that their computation complexity is cubic to the number of features which severely limits their applicability on high dimensional data, e.g., text data and genetic data.

Since the most discriminative information for feature selection is usually encoded in labels, it is very important to predict a good cluster indicators as pseudo labels for unsupervised feature selection [6]. Another important factor which effects the performance of feature selection is the consideration of outliers and noise [21]. Real data is usually not ideally distributed, outliers and noise often appear in the data, thus it is important or even necessary to consider robustness for unsupervised feature selection.

In light of all these factors, we propose a new unsupervised feature selection algorithm, i.e., Robust Unsupervised Feature Selection (RUFS). We perform robust clustering and robust feature selection simultaneously to select the most important and discriminative features for unsupervised learning. Specifically, cluster indicators are generated by local learning regularized robust non-negative matrix factorization, which is also a novel robust clustering method itself but we focus on unsupervised feature selection in this paper. Local learning [46] is used in robust clustering which incorporates both discriminative information and the geometric structure and is good at clustering data on manifold. We impose an orthogonal constraint on the cluster indicator matrix to ensure that the learned cluster indicators are much closer to the true cluster labels. We then simultaneously perform robust feature selection using learned cluster indicators. RUFS exploits the discriminative information and feature correlation in a joint framework. Aiming at feature selection, joint $l_{2,1}$ norms minimization is utilized to learn a robust feature selection matrix which is sparse in rows. In order for the proposed RUFS be practical for real world applications, we present a (projected) limited-memory BFGS based iterative algorithm to solve the optimization problem of RUFS. Experiments on different real world datasets show that the RUFS outperforms the state-of-the-arts.

4.2 Local Learning Regularization

According to [66], searching a good predictor f in a global way might not be a good strategy because the function set $f(\mathbf{x})$ may not contain a good predictor for the entire input space. However, it is much easier to produce good predictions on some local regions of the input space and it is usually more effective to minimize prediction cost for each region. [46] introduced a good way to construct

the local predictors, and we will use it as our local learning regularization term.

Denoting $\mathcal{N}(\mathbf{x}_i)$ as the neighborhood of $\mathbf{x}_i$, the local learning regularization aims to minimize the sum of prediction errors between the local prediction from $\mathcal{N}(\mathbf{x}_i)$ and the cluster assignment of $\mathbf{x}_i$:

$$\begin{aligned} J &= \sum_{k=1}^{K}\sum_{i=1}^{n}\left\|f_i^k(\mathbf{x}_i) - g_{ik}\right\| \\ &= \sum_{k=1}^{K}\sum_{i=1}^{n}\left\|\mathbf{k}_i^T(\mathbf{K}_i + n_i\lambda\mathbf{I})^{-1}\mathbf{g}_i^k - g_{ik}\right\| \\ &= \sum_{k=1}^{K}\sum_{i=1}^{n}\left\|\alpha_i^T\mathbf{g}_i^k - g_{ik}\right\| \\ &= \mathrm{Tr}\left[\mathbf{G}^T\mathbf{L}\mathbf{G}\right] \end{aligned} \tag{4.1}$$

where $f_i^k(\mathbf{x}_i)$ is the locally predicted label for k-th cluster from $\mathcal{N}(\mathbf{x}_i)$, λ is a positive parameter, $\mathbf{K}_i$ is the kernel matrix defined on the neighborhood of $\mathbf{x}_i$, i.e., $\mathcal{N}(\mathbf{x}_i)$, with size of n_i, $\mathbf{k}_i$ is the kernel vector defined between $\mathbf{x}_i$ and $\mathcal{N}(\mathbf{x}_i)$, $\mathbf{g}_i^k$ is the cluster assignments of $\mathcal{N}(\mathbf{x}_i)$, $\mathbf{L} = (\mathbf{A}-\mathbf{I})^{\mathbf{T}}(\mathbf{A}-\mathbf{I})$, $\mathbf{I} \in \mathcal{R}^{n\times n}$ is an identity matrix, and $\mathbf{A} \in \mathcal{R}^{n\times n}$ is defined by

$$\mathbf{A}_{ij} = \begin{cases} \alpha_{ij}, & \text{if } \mathbf{x}_j \in \mathcal{N}(\mathbf{x}_i) \\ 0, & \text{otherwise} \end{cases}. \tag{4.2}$$

4.3 The Objective Function

In this section, we present the objective function of the proposed Robust Unsupervised Feature Selection (RUFS) algorithm. To select discriminative features for unsupervised learning, learning accurate pseudo cluster labels is very important. NDFS [6] uses spectral clustering to predict the labels. In this work, however, we propose to utilize local learning regularized robust nonnegative matrix factorization with orthogonal constraint to learn the pseudo cluster labels. The reason is three-fold. First, since spectral clustering relies on similarity matrix computed from the original feature space. Thus unrelated or noisy features will have an adverse effect on clustering and henceforth hurt feature selection performance. Although nonnegative matrix factorization [67] decomposes the data matrix in the original feature space, the adverse effect will be mainly accumulated

in the learned cluster centers and won't hurt the indicators severely. Second, a robust clustering algorithm that considers outliers and noise usually improves the clustering performance [62]. Third, researchers [66][68][46] have shown that local learning is more effective than learning a good predictor in a global way since the function set may not contain a good predictor for the entire input space. Thus we use the local learning regularization to encode the discriminative information and the geometric structure via local predictors, which results in good clustering performance particularly on data embedded on manifold.

Our proposed robust clustering via local learning differs from [62] in that local learning is involved during the clustering procedure and an orthonormal constraint is imposed on the indicator matrix so that arbitrary scaling and trivial solutions could be avoided and more ideal pseudo cluster labels could be learned.

Given the proposed robust clustering with local learning, RUFS aims to solve the following optimization problem:

$$\begin{aligned} \min_{\mathbf{F},\mathbf{G},\mathbf{W}} \quad & \|\mathbf{X}-\mathbf{G}\mathbf{F}\|_{2,1}+\nu \mathrm{Tr}\left[\mathbf{G}^T\mathbf{L}\mathbf{G}\right]+\alpha\|\mathbf{X}\mathbf{W}-\mathbf{G}\|_{2,1}+\beta\|\mathbf{W}\|_{2,1} \\ \text{s.t.} \quad & \mathbf{G}\in\mathcal{R}_+^{n\times c}, \mathbf{G}=\mathbf{Y}\left(\mathbf{Y}^T\mathbf{Y}\right)^{-\frac{1}{2}}, \mathbf{F}\in\mathcal{R}_+^{c\times d}, \end{aligned} \tag{4.3}$$

where $\nu,\alpha,\beta\in\mathcal{R}_+$ are parameters. Robust feature selection is performed through jointly minimizing the last two terms (joint $l_{2,1}$ norms minimization), which is able to handle outliers and noise in data. The $l_{2,1}$ norm imposed on the feature selection matrix $\mathbf{W}$ guarantees the property of sparseness in rows. More specifically, $\mathbf{w}^j$ shrinks to zero if the j-th feature is less correlated to the pseudo labels $\mathbf{G}$. We can thus filter out the features corresponding to zero rows of $\mathbf{W}$ when performing feature selection.

Since $\mathbf{Y}$ by definition is a 0 or 1 matrix, the optimization of Eq. (4.3) is an NP-hard problem [69]. A commonly used strategy is to relax it to continuous values while keeping the key property, we thus constrain $\mathbf{G}$ to be orthonormal by columns, and the original optimization problem is relaxed

to

$$\begin{aligned} \min_{\mathbf{F},\mathbf{G},\mathbf{W}} \quad & \|\mathbf{X}-\mathbf{G}\mathbf{F}\|_{2,1} + \nu \mathrm{Tr}\left[\mathbf{G}^T\mathbf{L}\mathbf{G}\right] + \alpha\|\mathbf{X}\mathbf{W}-\mathbf{G}\|_{2,1} + \beta\|\mathbf{W}\|_{2,1} \\ \text{s.t.} \quad & \mathbf{G} \in \mathcal{R}_+^{n\times c}, \mathbf{G}^T\mathbf{G} = \mathbf{I}_c, \\ & \mathbf{F} \in \mathcal{R}_+^{c\times d}, \mathbf{W} \in \mathcal{R}^{d\times c}, \end{aligned} \tag{4.4}$$

where the first two terms learn the pseudo cluster labels using robust orthogonal nonnegative matrix factorization via local learning regularization while the last two terms simultaneously learn the feature selection matrix by joint $l_{2,1}$ norms minimization.

By solving optimization problem (4.4), we learn three components of the robust unsupervised feature selection model, i.e., the pseudo cluster labels $\mathbf{G}$ which is very close to the ideal scaled label indicators, the cluster centers $\mathbf{F}$ in the original whole feature space, and the feature selection matrix (or projection matrix for regression) $\mathbf{W}$ which is sparse in rows.

4.4 Optimization Algorithm

In the era of big data, high dimensional data is prevalent and the number of features is usually very high (otherwise, we may not need feature selection), for example, text data, genetic data, or image data with high resolution. In such cases, both UDFS and NDFS will be prohibitively slow since they share the computation complexity of $O\left(d^3\right)$ and memory complexity of $O\left(d^2\right)$. For practical use of unsupervised feature selection, we require algorithms to be able to handle large number of features and large number of data examples which are not only computationally efficient but also save memory.

Limited-memory quasi-Newton methods [70] are among the best candidates for solving large scale optimization problems when Hessian matrices cannot be computed at a reasonable cost or are not sparse. These methods maintain simple and compact approximations of Hessian matrices using only a few vectors that represent the approximations implicitly. Despite these modest storage requirements, they often yield an acceptable (albeit linear) rate of convergence. In this section, we present an iterative algorithm to efficiently solve Eq. (4.4) using L-BFGS [71] and projected

L-BFGS [72] methods.

To solve RUFS, we first rewrite the optimization problem as follows

$$\begin{aligned} \min_{\mathbf{F},\mathbf{G},\mathbf{W}} \quad & \|\mathbf{X}-\mathbf{G}\mathbf{F}\|_{2,1}+\nu\mathrm{Tr}\left[\mathbf{G}^T\mathbf{L}\mathbf{G}\right]+\beta\|\mathbf{W}\|_{2,1}+\alpha\|\mathbf{X}\mathbf{W}-\mathbf{G}\|_{2,1}+\frac{\zeta}{4}\left\|\mathbf{G}^T\mathbf{G}-\mathbf{I}_c\right\|_F^2 \\ \text{s.t.} \quad & \mathbf{G}\in\mathcal{R}_+^{n\times c},\mathbf{F}\in\mathcal{R}_+^{c\times d},\mathbf{W}\in\mathcal{R}^{d\times c} \end{aligned} \tag{4.5}$$

where ζ is a parameter to control the orthogonality condition. In practice, ζ should be large enough to insure the orthogonality satisfied. We first define the objective function

$$\mathscr{L}\left(\mathbf{G},\mathbf{F},\mathbf{W}\right)=\|\mathbf{X}-\mathbf{G}\mathbf{F}\|_{2,1}+\nu\mathrm{Tr}\left[\mathbf{G}^T\mathbf{L}\mathbf{G}\right]+\alpha\|\mathbf{X}\mathbf{W}-\mathbf{G}\|_{2,1}+\beta\|\mathbf{W}\|_{2,1}+\frac{\zeta}{4}\left\|\mathbf{G}^T\mathbf{G}-\mathbf{I}_c\right\|_F^2, \tag{4.6}$$

denoting

$$\begin{aligned} \mathbf{r}_1 &= \left[\left\|\mathbf{x}^1-\mathbf{g}^1\mathbf{F}\right\|_2,\ldots,\left\|\mathbf{x}^n-\mathbf{g}^n\mathbf{F}\right\|_2\right]^T, \\ \mathbf{r}_2 &= \left[\left\|\mathbf{x}^1\mathbf{W}-\mathbf{g}^1\right\|_2,\ldots,\left\|\mathbf{x}^n\mathbf{W}-\mathbf{g}^n\right\|_2\right]^T, \\ \mathbf{r}_3 &= \left[\left\|\mathbf{w}^1\right\|_2,\ldots,\left\|\mathbf{w}^d\right\|_2\right]^T, \end{aligned}$$

the partial derivatives of $\mathscr{L}\left(\mathbf{G},\mathbf{F},\mathbf{W}\right)$ w.r.t. $\mathbf{G},\mathbf{F},$ and $\mathbf{W}$ can be obtained

$$\begin{aligned} \nabla_{\mathbf{G}}\mathscr{L} &= \left(\mathbf{G}\mathbf{F}-\mathbf{X}\right)\mathbf{F}^T\oslash\left[\mathbf{r}_1\otimes\mathbb{1}_{1\times c}\right]+2\nu\mathbf{L}\mathbf{G}+\alpha\left(\mathbf{G}-\mathbf{X}\mathbf{W}\right)\oslash\left[\mathbf{r}_2\otimes\mathbb{1}_{1\times c}\right]+\zeta\mathbf{G}\left(\mathbf{G}^T\mathbf{G}-\mathbf{I}_c\right), \\ \nabla_{\mathbf{F}}\mathscr{L} &= \mathbf{G}^T\left[\left(\mathbf{G}\mathbf{F}-\mathbf{X}\right)\oslash\left[\mathbf{r}_1\otimes\mathbb{1}_{1\times d}\right]\right], \\ \nabla_{\mathbf{W}}\mathscr{L} &= \alpha\mathbf{X}^T\left[\left(\mathbf{X}\mathbf{W}-\mathbf{G}\right)\oslash\left[\mathbf{r}_2\otimes\mathbb{1}_{1\times c}\right]\right]+\beta\mathbf{W}\oslash\left[\mathbf{r}_3\otimes\mathbb{1}_{1\times c}\right], \end{aligned}$$

where $\otimes$ is the Kronecker product, $\oslash$ is the element-wise division, and $\mathbb{1}$ is an all 1 matrix. Solutions

of problem (4.5) satisfy the Kuhn-Tucker conditions

$$\begin{cases} \frac{\partial \mathscr{L}}{\partial G_{ik}} = 0 \text{ if } G_{ik} > 0; \frac{\partial \mathscr{L}}{\partial G_{ik}} \geq 0 \text{ if } G_{ik} = 0 \\ \\ \frac{\partial \mathscr{L}}{\partial F_{kj}} = 0 \text{ if } F_{kj} > 0; \frac{\partial \mathscr{L}}{\partial F_{kj}} \geq 0 \text{ if } F_{kj} = 0 \\ \\ \frac{\partial \mathscr{L}}{\partial W_{jk}} = 0 \end{cases} . \tag{4.7}$$

The projection operator

$$[T_{\Omega}\mathbf{M}]_{ij} = \begin{cases} M_{ij} & \text{if} \quad X_{ij} > 0 \\ \min\{M_{ij}, 0\} & \text{if} \quad X_{ij} = 0 \end{cases} \tag{4.8}$$

can be helpful because $(\mathbf{G}^*, \mathbf{F}^*, \mathbf{W}^*)$ is a solution of problem (4.5) if and only if

$$\left(T_{\mathcal{R}_+^{n\times c}} \nabla \mathbf{G}^*, T_{\mathcal{R}_+^{c\times d}} \nabla \mathbf{F}^*, \nabla \mathbf{W}^*\right) = \mathbf{0}. \tag{4.9}$$

Given a tolerance τ, an approximate solution to problem (4.5) is any matrix triplet $(\mathbf{G}, \mathbf{F}, \mathbf{W})$ such that

$$\left\| \left(T_{\mathcal{R}_+^{n\times c}} \nabla \mathbf{G}, T_{\mathcal{R}_+^{c\times d}} \nabla \mathbf{F}, \nabla \mathbf{W}\right) \right\| \leq \tau. \tag{4.10}$$

In next subsection, we will present a limited-memory BFGS based alternating iterative algorithm to efficiently solve problem (4.5).

4.4.1 Limited-memory BFGS

Recall that each step of the BFGS method has the form

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \mathbf{H}_k \nabla f_k, \tag{4.11}$$

where α_k is the step length and $\mathbf{H}_k$ is the inverse Hessian approximation. Since $\mathbf{H}_k$ will generally be dense, the cost of storing and manipulating it is prohibitive when the number of variables is large. To circumvent this problem, limited-memory BFGS computes a modified version of $\mathbf{H}_k$

implicitly by storing a certain number (say, m) of most recent correction pairs $\{\mathbf{s}_i, \mathbf{y}_i\}$ using a two-loop recursive procedure to compute the product $\mathbf{H}_k \nabla f$ efficiently [70] shown in Algorithm 1. The limited-memory BFGS algorithm can thus be stated formally shown in Algorithm 2.

Algorithm 1 L-BFGS two-loop recursion

$\mathbf{H}_k^0 = \frac{\mathbf{s}_{k-1}^T \mathbf{y}_{k-1}}{\mathbf{y}_{k-1}^T \mathbf{y}_{k-1}} \mathbf{I}$
$\mathbf{q} \leftarrow \nabla f_k$
for $i = k-1, k-2, \ldots, k-m$ **do**
  $\alpha_i \leftarrow \rho_i \mathbf{s}_i^T \mathbf{q}$
  $\mathbf{q} \leftarrow \mathbf{q} - \alpha_i \mathbf{y}_i$
end for
$\mathbf{r} \leftarrow \mathbf{H}_k^0 \mathbf{q}$
for $i = k-m, k-m+1, \ldots, k-1$ **do**
  $\beta \leftarrow \rho_i \mathbf{y}_i^T \mathbf{r}$
  $\mathbf{r} \leftarrow \mathbf{r} + \mathbf{s}_i (\alpha_i - \beta)$
end for
return $\mathbf{H}_k \nabla f_k = \mathbf{r}$

Algorithm 2 L-BFGS

Input: Staring point $\mathbf{x}_0$ and an integer $m > 0$
$k \leftarrow 0$
repeat
  Compute $\mathbf{p}_k \leftarrow -\mathbf{H}_k \nabla f_k$ using a two-loop recursion
  Compute $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k + \alpha_k \mathbf{p}_k$, where α_k is chosen to
    satisfy the Wolfe conditions
  if $k > m$ **then**
    Discard the vector pair $\{\mathbf{s}_{k-m}, \mathbf{y}_{k-m}\}$
  end if
  Save $\mathbf{s}_k \leftarrow \mathbf{x}_{k+1} - \mathbf{x}_k, \mathbf{y}_k \leftarrow \nabla f_{k+1} - \nabla f_k$
  $k \leftarrow k + 1$
until $\|\nabla f_k\| \leq \tau$
Output: $\mathbf{x}_k$

When nonnegative constraints are imposed, a projected version of limited-memory BFGS algorithm is required. There are various projected limited-memory BFGS algorithms, here we choose BMLVM algorithm [72] for its faster speed than L-BFGS-B [73] and ease of use.

4.4.2 RUFS Algorithm

We adopt an alternating optimization (AO) strategy to solve RUFS and list it in Algorithm 4. Following the convergence analysis for a general AO approach, the convergence of Algorithm 4 can

Algorithm 3 BMLVM

Input: Staring point $\mathbf{x}_0$ and an integer $m > 0$
$k \leftarrow 0$
repeat
  Compute $\mathbf{p}_k \leftarrow -\mathbf{H}_k \nabla f_k$ using a two-loop recursion
  if $\langle T_\Omega(\mathbf{H}_k \nabla f_k), \nabla f_k \rangle > 0$ **then**
    $\mathbf{p}_k \leftarrow -T_\Omega(\mathbf{H}_k \nabla f_k)$
  else
    $\mathbf{p}_k \leftarrow -T_\Omega \nabla f_k$
  end if
  Compute $\mathbf{x}_{k+1} \leftarrow [\mathbf{x}_k + \alpha_k \mathbf{p}_k]_+$, where α_k is chosen
    to satisfy the Wolfe conditions
  if $k > m$ **then**
    Discard the vector pair $\{\mathbf{s}_{k-m}, \mathbf{y}_{k-m}\}$
  end if
  Save $\mathbf{s}_k \leftarrow \mathbf{x}_{k+1} - \mathbf{x}_k, \mathbf{y}_k \leftarrow T_\Omega \nabla f_{k+1} - T_\Omega \nabla f_k$
  $k \leftarrow k + 1$
until $\|T_\Omega \nabla f_k\| \leq \tau$
Output: $\mathbf{x}_k$

Algorithm 4 RUFS

Input: $\mathbf{X} \in \mathcal{R}^{n \times d}, \nu, \alpha, \beta, c,$ and p
  Construct $\mathbf{L}$ from Eq. (4.1)
  Initialize $\mathbf{G}_0$ (e.g., by K-means)
  Initialize $\mathbf{F}_0 \leftarrow \left[\left(\mathbf{G}^T\mathbf{G}\right)^{-1}\mathbf{G}^T\mathbf{X}\right]_+$
  Initialize $\mathbf{W}_0$
  $k \leftarrow 0$
  repeat
    Fixing $\mathbf{G}_k$, compute $\mathbf{W}_{k+1}$ from
      Algorithm 2 given $\mathbf{G}_k, \mathbf{W}_k, \alpha, \beta$
    Fixing $\mathbf{F}_k$ and $\mathbf{W}_{k+1}$, compute $\mathbf{G}_{k+1}$ from
      Algorithm 3 given $\mathbf{G}_k, \mathbf{F}_k, \mathbf{W}_{k+1}\mathbf{L}, \nu,$ and α
    Fixing $\mathbf{G}_{k+1}$, compute $\mathbf{F}_{k+1}$ from
      Algorithm 3 given $\mathbf{G}_{k+1}, \mathbf{F}_k$
    $k \leftarrow k + 1$
  until $\left\| \left(T_{\mathcal{R}_+^{n \times c}} \nabla \mathbf{G}_k, T_{\mathcal{R}_+^{c \times d}} \nabla \mathbf{F}_k, \nabla \mathbf{W}_k \right) \right\| \leq \tau$
Output: Sort all d features according to $\|\mathbf{w}_k^j\|_2$ in descending order and select the top p ranked features.

Table 4.1: Dataset Description.

Dataset	# of Samples	# of Features	# of Classes
ORL	400	1024	40
COIL20	1440	1024	20
BinaryAlphadigits	1404	320	36
UMIST	575	644	20
Isolet	1560	617	26
WebKB4	4199	1000	4

be shown to be locally and q-linearly convergent [74].

4.4.3 Complexity Analysis

The two-loop recursion scheme requires $O(4tmdc)$ for computing $\mathbf{W}$ and $\mathbf{F}$ and $O(4tmnc)$ for computing $\mathbf{G}$, we thus have $O(4tm(2d+n)c)$ scaler multiplications where t is the total number of inner iterations of Algorithm 4. Computing partial gradients w.r.t. $\mathbf{G}$, $\mathbf{F}$, and $\mathbf{W}$ are $t_G * O(3ndc + cnk_{nn})$, $t_F * O(2ndc)$, and $t_W * O(2ndc)$ respectively, where t_A is the total number of inner iterations of Algorithm 4 computing matrix $\mathbf{A}$. Evaluation of objective function values requires about $\#lineSearchIter * O(cnk_{nn} + ndc + nc^2)$, where k_{nn} is the number of nearest neighbors when constructing the sparse adjacency matrix for computing the local learning regularization matrix. The computation of projection operation can be ignored compared to the computation of gradients and objective function values because only boolean operations are performed. The memory complexity of RUFS is $O(nk_{nn} + nd + nc + dc)$. Note that both UDFS and NDFS require $O(d^2) + O(nk_{nn}) + O(cn)$ memory cost and $O(d^3) + O(cnk_{nn})$ computation complexity, which will be prohibitively slow when the original feature size d is very large.

Since the computational complexity and memory cost of RUFS is linear to the feature size d and the data size n, the proposed method can be run on big data. The only restriction is the requirement that data and intermediate matrices should be stored in memory since it is a sequential and iterative algorithm. In this case, one can use Apache Spark[1] to process big data as it supports cyclic data flow and in-memory computing.

[1] http://spark.apache.org/

4.5 Experiments

In this section, we conduct experiments to evaluate RUFS. Following previous unsupervised feature selection work [4][5][6], we only evaluate the performance of RUFS for feature selection on clustering due to space limit.

4.5.1 Datasets

The evaluation is performed on 6 benchmark real world datasets including ORL (AT&T)[2], COIL20[3], Binary Alphadigits[4], UMIST[5], Isolet1[6], and WebKB4 [46]. Detailed information is summarized in Table 4.1.

4.5.2 Visualization of Selected Features by RUFS

As an illustration, we show top 80 features selected by RUFS on the ORL data set in Figure 4.1. We see that pixels around the eyes, nose, and lips are selected. Actually these features are also important for a human to recognize face. From the figure, we see that unsupervised feature indeed is capable of selecting important features for representing an object.

4.5.3 RUFS Helps Classification

One important question is if unsupervised feature selection can improve classification, which usually required label information for training. We found that RUFS can further improve classification accuracy. For all six real world benchmark datasets, we randomly selected half for training and the remaining half for test. We did unsupervised feature selection by RUFS with $\alpha = 10$, $\beta = 10$, and $\nu = 10$ on all data examples ignoring the labels and plot the accuracy and running time of multi-class logistic regression using different number of top features selected by RUFS on all six real world benchmark datasets in figure 4.2. The figure shows that RUFS improves the classification accuracy and usually reduces the running time for training the classifiers on all six

[2]http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html
[3]http://www.cad.zju.edu.cn/home/dengcai/Data/MLData.html
[4]http://www.cs.nyu.edu/~roweis/data.html
[5]http://www.sheffield.ac.uk/eee/research/iel/research/face
[6]http://archive.ics.uci.edu/ml/datasets/ISOLET

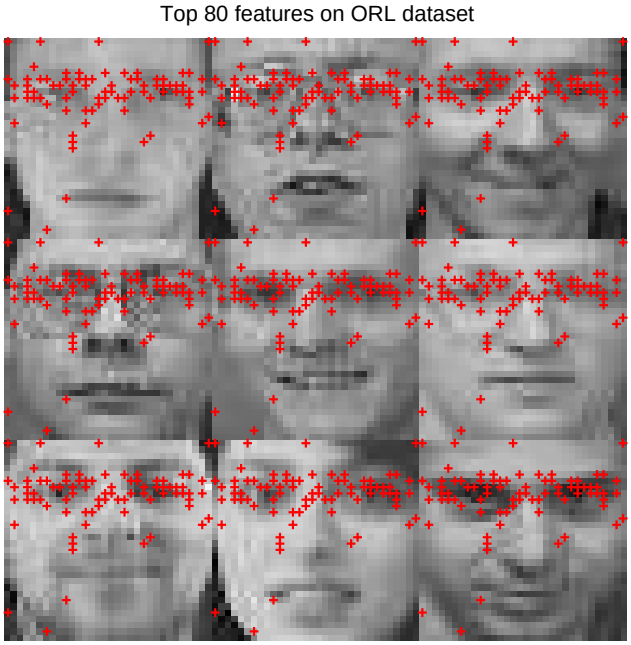


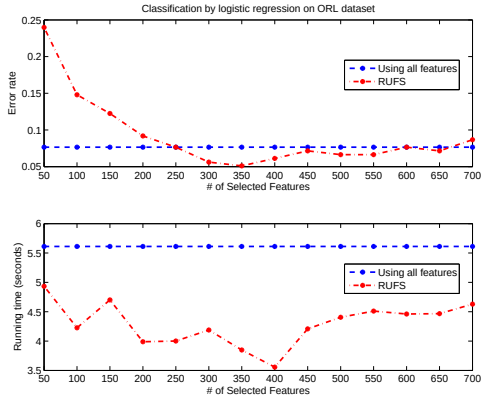
Figure 4.1: Top 80 selected features by RUFS on ORL dataset.

real world benchmark datasets. The lowest error rate could be achieved with an appropriate number of feature chosen between 50 and the feature size. Too few features gives a very poor accuracy because not all important features are selected. On the opposite side, too many features also leads to sub-minimal error rate because irrelevant or noisy features are selected which undermines the classifier.

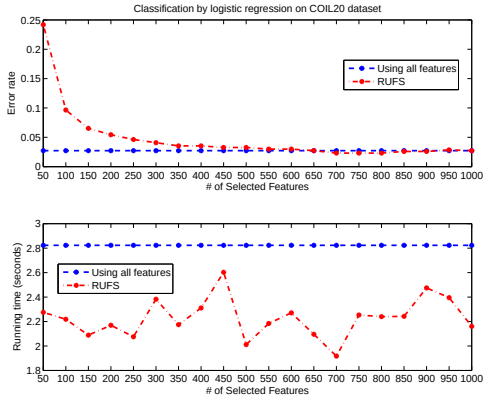
4.5.4 Compared Methods

We compare RUFS with the following unsupervised feature selection algorithms.

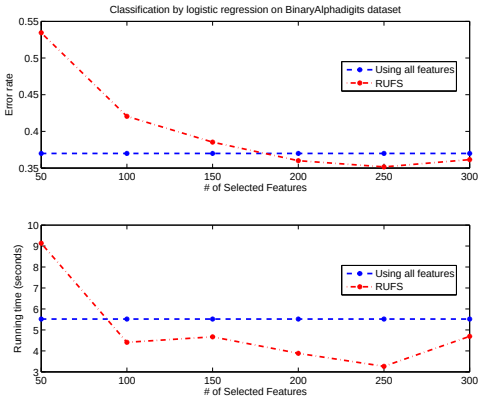
1. **Baseline**: All original features are adopted.
2. **LS**: Laplacian Score [11] which selects features that best preserve the local manifold structure.
3. **MCFS**: Mutli-Cluster Feature Selection [4] where features are selected using spectral regression with l_1-norm regularization.
4. **UDFS**: Unsupervised Discriminative Feature Selection [5] which exploits local discriminative



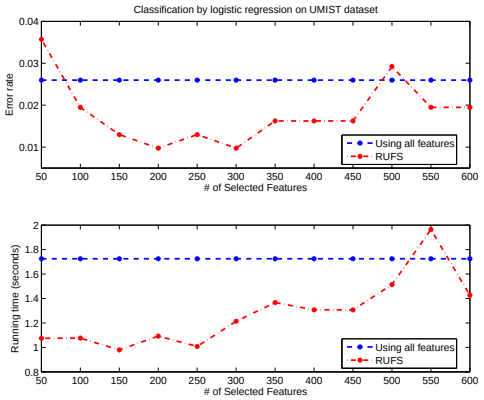
(a) ORL



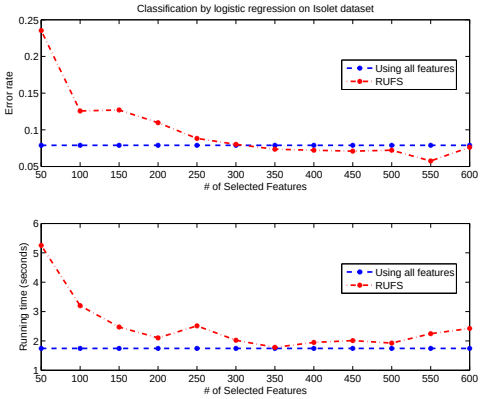
(b) COIL20



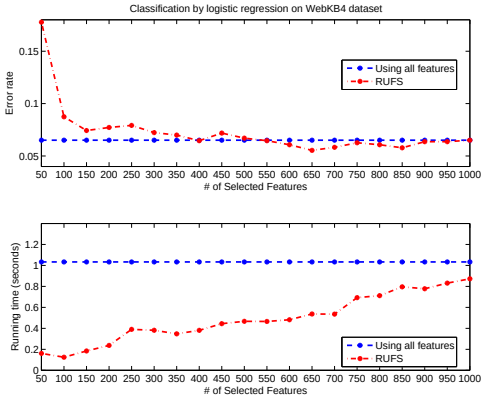
(c) BinaryAlphadigits



(d) UMIST



(e) Isolet



(f) WebKB4

Figure 4.2: Classification error rate under different number of top selected features by multi-class logistic regression on all six real world benchmark datasets.

Table 4.2: Clustering Results (ACC% $\pm$ std) of Different Feature Selection Algorithms.

Dataset	All Features	Laplacian Score	MCFS	UDFS	NDFS	RUFS
ORL	51.1 ± 3.2	47.2 ± 2.5	49.7 ± 3.7	51.3 ± 3.0	52.3 ± 3.2	$\mathbf{53.4 \pm 3.8}$
COIL20	60.4 ± 4.5	56.4 ± 4.6	60.9 ± 4.7	59.8 ± 4.4	59.7 ± 3.3	$\mathbf{62.0 \pm 3.2}$
BinaryAlphadigits	41.0 ± 2.1	42.3 ± 1.8	41.8 ± 2.3	42.4 ± 1.8	42.4 ± 1.7	$\mathbf{42.7 \pm 1.7}$
UMIST	41.7 ± 2.5	44.1 ± 2.7	45.4 ± 2.6	45.3 ± 2.4	48.2 ± 3.6	$\mathbf{49.1 \pm 3.2}$
Isolet	59.7 ± 3.6	56.2 ± 3.7	56.9 ± 4.7	56.2 ± 3.8	63.0 ± 4.6	$\mathbf{64.5 \pm 3.2}$
WebKB4	69.2 ± 8.6	49.1 ± 7.9	59.5 ± 9.6	60.1 ± 5.8	69.2 ± 6.7	$\mathbf{74.2 \pm 2.5}$

Table 4.3: Clustering Results (NMI% $\pm$ std) of Different Feature Selection Algorithms.

Dataset	All Features	Laplacian Score	MCFS	UDFS	NDFS	RUFS
ORL	74.0 ± 1.9	71.5 ± 1.2	73.7 ± 1.8	73.4 ± 1.6	74.9 ± 1.9	$\mathbf{75.1 \pm 1.8}$
COIL20	76.3 ± 1.8	71.8 ± 2.0	74.9 ± 2.2	74.7 ± 1.6	76.0 ± 1.6	$\mathbf{77.0 \pm 2.2}$
BinaryAlphadigits	57.6 ± 1.3	58.5 ± 0.9	58.3 ± 1.2	58.8 ± 0.9	58.6 ± 0.8	$\mathbf{59.4 \pm 1.0}$
UMIST	63.9 ± 1.8	65.9 ± 1.4	66.6 ± 1.7	65.2 ± 1.6	66.5 ± 2.2	$\mathbf{68.8 \pm 2.4}$
Isolet	75.9 ± 1.6	73.1 ± 1.5	73.1 ± 1.4	72.8 ± 1.8	78.6 ± 1.6	$\mathbf{78.9 \pm 1.1}$
WebKB4	46.7 ± 3.1	29.2 ± 11.5	37.4 ± 15.3	34.5 ± 5.2	45.3 ± 4.9	$\mathbf{49.5 \pm 2.9}$

information and feature correlations simultaneously and considers the manifold structure as well.

5. **NDFS**: Nonnegative Discriminative Feature Selection [6] where features are selected by a joint framework of nonnegative spectral analysis and $l_{2,1}$-norm regularized regression.

4.5.5 Experiment Setup

Following previous work, two widely used evaluation metrics, i.e., Accuracy (ACC) and Normalized Mutual Information (NMI) are used in this chapter.

There are some parameters to be set. Following previous work, for LS, MCFS, UDFS, NDGS, and RUFS, we fix $k = 5$ for all the datasets to specify the neighborhood size. To fairly compare different unsupervised feature selection methods, we tune the parameters for all methods by a "grid-search" strategy from $\{10^{-6}, 10^{-4}, \ldots, 10^{4}, 10^{6}\}$. The number of selected features are set as $\{50, 100, 150, \ldots 300\}$ for all datasets. Best clustering results from the optimal parameters are reported for all the algorithms. In the evaluation, we use K-means to cluster samples based on the selected features. Since K-means depends on initialization, following previous work, we repeat clustering 20 times with random initialization for each setup. The average results with standard deviation are reported.

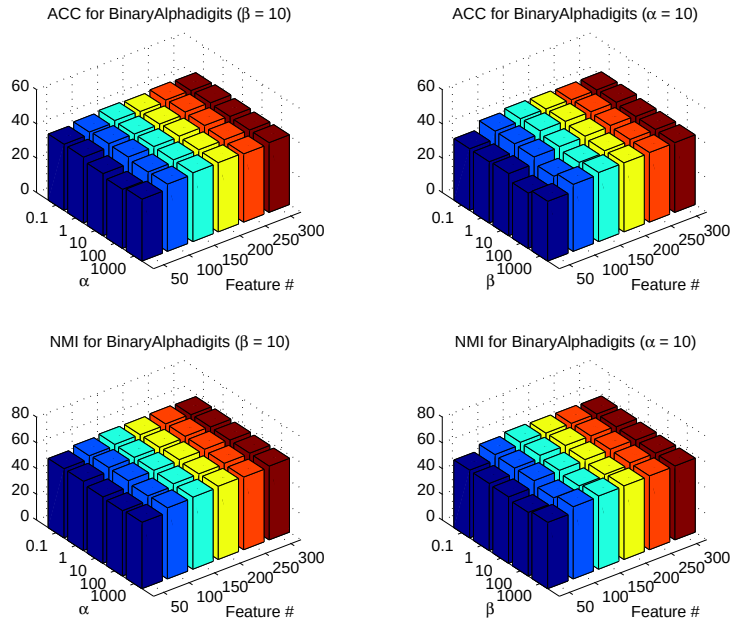


Figure 4.3: ACC and NMI of RUFS with different α, β and feature numbers while keeping $\nu = 10$.

Table 4.4: Average Running Time (seconds).

Dataset	UDFS	NDFS	RUFS
COIL20	42.4	50.4	**32.2**
WebKB4	112.9	281.3	**86.1**

4.5.6 Results and Discussion

We list the experimental results of different methods in Table 4.2 and Table 4.3. We observe from the clustering results that feature selection is important and effective. Not only can number of features be significantly reduced which makes posterior processing more efficient, but clustering performance can also be greatly improved. A new observation is that robust analysis is important for unsupervised learning. Consideration of outliers and noise usually improves the performance of feature selection, which has also been observed in supervised scenario. At last, RUFS achieves the best performance. This can be mainly explained by the following reasons. First, joint learning is performed between robust label learning and robust feature selection. Second, local learning is exploited which results in more accurate pseudo labels. Third, outliers and noise are considered during processes of both label learning and feature learning, so that more accurate and discriminative pseudo labels can be obtained.

We also study the sensitiveness of parameters. We only report the results on BinaryAlphadigits dataset with $\nu = 10$ (sensitiveness under other values of ν is similar) on Figure 4.3. The experimental results show that our method is not very sensitive to α and β with wide ranges. However, the performance is relatively sensitive to the number of selected features, which is still an open problem. For practitioners, we suggest using a validation set with ground truth under an affordable cost to tune the parameters by e.g. grid search. Also different users may label the data points differently, we can group the users by their ways of labeling and ask similar users to construct the validation set to tune parameters that work best for them.

We finally compare the running time of UDFS, NDFS, and RUFS in Table 4.4 on COIL20 and WebKB4 datasets (other datasets have either a smaller sample size or a smaller feature size). The calculations are performed using an Intel(R) Core(TM) i7 CPU M620 @ 2.67GHz with 4.00GB memory and 64-bit Windows 7 operating system. The empirical results in Table 4.4 are consistent with the theoretical analysis.

4.5.7 Limitation

Please be noted that the reason why the small-scale public benchmark datasets are used in the experiments is because researchers in the literature usually compare algorithms on these datasets. One advantage of using these datasets is that it is easier for people to assess and compare different methods. Another advantage is that it doesn't require expensive distributive system for evaluation, which would allow more researchers to test the algorithms. However, since the experiments are conduct on small-scale datasets, the algorithms' performance rank might not be exactly the same on large scale datasets. But it can be qualitatively argued that the proposed method would still be superior to the baseline methods because the distribution of outliers on large scale datasets would be similar if it is not exactly the same to these small-scale benchmark real world datasets. For example, the distribution of outliers of a large scale ORL dataset will be highly expected to be similar to the small scale ORL dataset used in this experiment. For quantitative validation, experiments on large-scale datasets need to be done before conclusions could be made on large-scale datasets, which is a promising research direction for future work.

4.6 Summary

In this chapter, we propose a new robust unsupervised feature selection approach called RUFS, which jointly performs robust label learning via local learning regularized robust orthogonal non-negative matrix factorization and robust feature learning via joint $l_{2,1}$-norms minimization. To make RUFS be applicable for large scale feature selection tasks, we present a (projected) limited-memory based iterative algorithm to solve it. Experimental results on different real world datasets validate the effectiveness of the proposed method. The proposed method deals with single view data. In the next chapter, we will study how the choice of norms for data fitting and feature selection terms affects the ultimate unsupervised feature selection performance.

Chapter 5

Joint Adaptive Loss and l_2/l_0-norm Minimization for Unsupervised Feature Selection

In the last chapter, we studied unsupervised feature selection from single-view perspective. We've seen that the use of $l_{2,1}$-norm play an important role. In this chapter, we study how choice of norms for data fitting and feature selection affects the ultimate unsupervised feature selection performance.

5.1 Introduction

Theoretically speaking, understanding the fundamental aspects of unsupervised feature selection methods is important, because it would provide theoretical and empirical guidance on the design of unsupervised feature selection algorithms, this motivates us to study the desirable properties of models for unsupervised feature selection and propose a new model which meets all properties.

The state-of-the-art unsupervised feature selection methods have three defects. First, [5, 6, 75] use $l_{2,1}$-norm to select features. But one disadvantage of $l_{2,1}$-norm is that it over-penalizes large weights. As is known, important features usually have larger weights. Good feature selection methods try to get a trade-off between data approximation and joint sparsity of feature weights. However, over-penalizing features with large weights might hurt data approximation performance, and force the algorithm to deviate the learned weights from the true feature weights. An ideal feature selection function should have two properties. (1) It should have the sparsity-inducing property. (2) It should equally penalize large weights and small weights, leading to a fair competition between different features. We see that traditionally favored norms such as l_1-norm, $l_{2,1}$-norm, and $l_{\infty,1}$-norm are sparsity-inducing convex models that satisfy the first property but fail to comply with the second property.

Second, the data fitting term is also very important for feature selection for the feature weights are learned from minimizing regularized fitting error. A good fitting term should satisfy two

properties. (1) It should enable the fitting model to approximate normal examples as much as possible. (2) It should impose small penalty on large loss on outliers. However, the squared Frobenius norm [6] only satisfies the first property thus is not robust for outliers, whereas $l_{2,1}$-norm [75] only satisfies the second one, thus is sensitive to small loss, i.e., it penalizes more for small loss than squared Frobenius norm.

Based on the observation and reasoning mentioned above, we propose a new unsupervised feature selection method, i.e. Adaptive Unsupervised Feature Selection with explicit l_2/l_0-norm (AUFS). We propose to use l_2/l_0-norm (Rigorously, it is not a norm. People use norm for convenience in the literature.) to do feature selection instead of traditional $l_{2,1}$-norm and use the adaptive loss to penalize the approximation error instead of squared Frobenius norm and $l_{2,1}$-norm. The advantage of l_2/l_0-norm is that it not only has the sparsity-inducing property, but also equally penalizes large weights and small weights. Thus it leads to smaller approximate error and avoids forcing the algorithm to favor small weights. Adaptive loss could achieve a good balance between small loss on normal data examples and large loss on outliers. Actually, when the parameter changes from 0 to ∞, the adaptive loss term varies from $l_{2,1}$-norm to squared Frobenius norm. For label learning, we use nonnegative orthogonal constrained spectral clustering. We propose to directly solve the nonnegative orthogonal constrained optimization problem by applying the Lagrange multiplier theory. We derive and present an efficient iterative algorithm to solve the optimization problem of AUFS in terms of both computation complexity and memory cost. The algorithm is flexible and general in that traditional $l_{2,1}$-norm and $l_{\infty,1}$-norm can also be used in place of l_2/l_0-norm without affecting the convergence property, whereas the optimization algorithms in [5, 6, 75] can only deal with $l_{2,1}$-norm. Experiments on seven different real world data sets show that AUFS significantly outperforms the state-of-the-arts.

Note that although the idea of l_2/l_0-norm regularization appeared in [76], they proposed a general Lipschitz auxiliary function to solve the optimization problem and they applied the regularization on multi-task learning problems. [77] also proposed a l_2/l_0-norm constrained optimization problem to find a subset of features and learn a linear transformation to optimize the Locality Preserving Criterion based on these features. A variation of Alternating Direction Method is applied to solve the optimization problem. In our work we first use proximal gradient descent to solve the op-

timization problem and first applied it on unsupervised feature selection. This work improves one's understanding on fundamental aspects of unsupervised feature selection and provides theoretical and empirical guidance on the design of unsupervised feature selection algorithms.

5.2 l_2/l_0-norm and Adaptive Loss

For any matrix $\mathbf{M} \in \mathcal{R}^{r\times p}$, its $l_{2,1}$-norm is defined as $\|\mathbf{M}\|_{2,1} = \sum_{i=1}^{r} \sqrt{\sum_{j=1}^{p} m_{ij}^2} = \sum_{i=1}^{r} \left\|\mathbf{m}^i\right\|_2$, and its l_2/l_0-norm (or $l_{2,0}$-norm) is defined by $\|\mathbf{M}\|_{2,0} = \sum\limits_{i=1}^{r} \left\|\mathbf{m}^i\right\|_0$, where for a vector $\mathbf{x}$, $\|\mathbf{x}\|_0 = 1$ if $\mathbf{x} \neq \mathbf{0}$, $\|\mathbf{x}\|_0 = 0$ if $\mathbf{x} = \mathbf{0}$. To see how l_2/l_0-norm and traditional sparsity-inducing norms such as $l_{2,1}$-norm treat small and large weights differently, for a nonzero vector $\mathbf{x}$ s.t. $\|\mathbf{x}\|_2 = 1$ and $r > 0$, $\|r\mathbf{x}\|_{2,0} = 1$, but $\|r\mathbf{x}\|_{2,1} = r\|\mathbf{x}\|_{2,1} = r\|\mathbf{x}\|_2 = r$. Thus the penalty on a nonzero vector imposed by the $l_{2,1}$-norm is proportional to the norm of the vector whereas l_2/l_0-norm imposes a constant penalty on the vector.

Given a matrix $\mathbf{X}$, the adaptive loss function [78] is defined by $\|\mathbf{X}\|_\sigma \triangleq \sum_i \frac{(1+\sigma)\left\|\mathbf{x}^i\right\|_2^2}{\left\|\mathbf{x}^i\right\|_2+\sigma}$, $\sigma > 0$. The adaptive loss function has the following properties:

a $\|\mathbf{X}\|_\sigma$ is nonnegative and convex, which is desirable for a loss function.

b $\|\mathbf{X}\|_\sigma$ is twice differentiable, which is desirable for optimization.

c When$\forall i, \left\|\mathbf{x}^i\right\|_2^2 \ll \sigma$, then$\|\mathbf{X}\|_\sigma \to \frac{1+\sigma}{\sigma} \|\mathbf{X}\|_F^2$.

d When$\forall i, \left\|\mathbf{x}^i\right\|_2^2 \gg \sigma$, then$\|\mathbf{X}\|_\sigma \to (1+\sigma) \|\mathbf{X}\|_{2,1}$.

e When $\sigma \to 0$, then $\|\mathbf{X}\|_\sigma \to \|\mathbf{X}\|_{2,1}$.

f When $\sigma \to \infty$, then $\|\mathbf{X}\|_\sigma \to \|\mathbf{X}\|_F^2$.

To see why adaptive loss achieves a good balance between squared Frobenius norm and $l_{2,1}$-norm, for a row matrix $\mathbf{x}$,

$$\begin{aligned} \|\mathbf{x}\|_{2,1} - \|\mathbf{x}\|_\sigma &= \frac{\sigma\|\mathbf{x}\|_2\left(1-\|\mathbf{x}\|_2\right)}{\|\mathbf{x}\|_2+\sigma} > 0 \text{ if } 0 < \|\mathbf{x}\|_2 < 1, \\ \|\mathbf{x}\|_F^2 - \|\mathbf{x}\|_\sigma &= \frac{\|\mathbf{x}\|_2^2\left(\|\mathbf{x}\|_2-1\right)}{\|\mathbf{x}\|_2+\sigma} > 0 \text{ if } \|\mathbf{x}\|_2 > 1. \end{aligned}$$

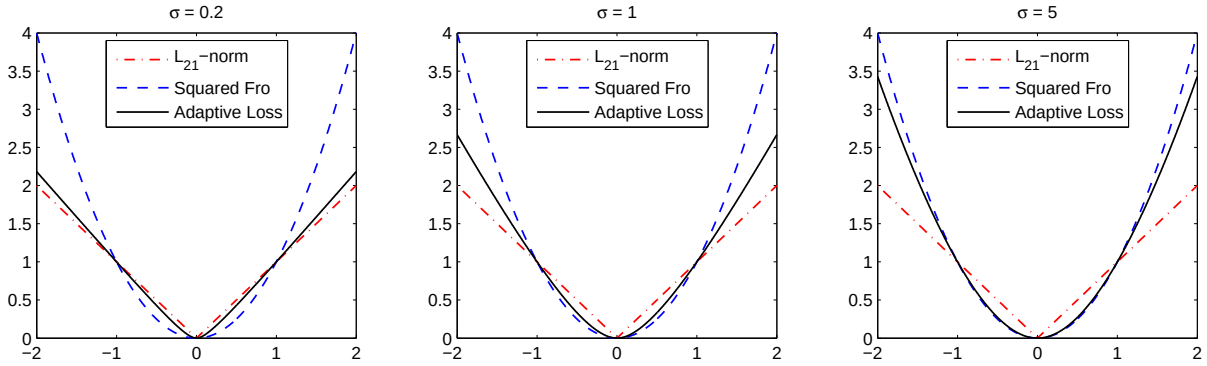


Figure 5.1: Illustration of $l_{2,1}$-norm, sqaured Frobenius norm and adaptive loss with different σ.

It is clear that adaptive loss imposes smaller penalty than squared Frobenius norm on large loss and smaller penalty than $l_{2,1}$-norm on small loss too. Consequently, it is insensitive to both outliers and small loss examples. Figure 5.1 illustrates $l_{2,1}$-norm, squared Frobenius norm and adaptive loss with different σ for a real scalar. It can be seen from Figure 5.1 that adaptive loss achieves a good balance between squared Frobenius norm and $l_{2,1}$-norm.

5.3 The Objective Function

As is mentioned in the introduction section, traditional unsupervised feature methods have three drawbacks. First, $l_{2,1}$-norm [6, 75] favors features with small weight and over-penalizes features with large weights. This preference will in turn hurt the data fitting performance. Thus, the algorithms using $l_{2,1}$-norm usually deviate the learned weight matrix from the true one. We thus propose to use l_2/l_0-norm as the feature selection regularization term. Not only is sparsity-inducing, but l_2/l_0-norm also equally penalizes all features. Thus it won't significantly hurt the fitting performance. If a feature is irrelevant or unimportant, l_2/l_0-norm can set the corresponding feature weight to zero. If a feature is relevant and important, l_2/l_0-norm equally penalizes them, thus the resulted feature weight matrix reflects the true degree of importance without prior preference. The second defect of traditional methods is that the fitting term over-penalizes either small loss or large loss. Squared Frobenius norm in [6] is sensitive to large loss and consequently is not robust for outliers, whereas $l_{2,1}$-norm in [75] is sensitive to small loss, i.e., it penalizes more for small loss than squared Frobenius norm. The ideal loss term should be not only robust to outliers but also insensitive to small loss.

Adaptive loss is an ideal loss term that achieves a good balance between the squared Frobenius norm and $l_{2,1}$-norm such that the convex and differentiable properties are satisfied. Third, in [6, 75], the orthogonal constraint is added to the objective function as an augmented regularization term. By setting a very large parameter (e.g., 10^8), the solution doesn't violate the orthogonal constraint too much. Here we propose to directly solve the nonnegative orthogonal constraint optimization problem by applying the Lagrangian multiplier theory.

AUFS solves the following optimization problem:

$$\begin{aligned} \min \quad & \mathrm{Tr}\left[\mathbf{Y}^T\mathbf{L}\mathbf{Y}\right] + \lambda\|\mathbf{X}\mathbf{W} - \mathbf{Y}\|_\sigma + \nu\|\mathbf{W}\|_{2,0} \\ \text{s.t.} \quad & \mathbf{Y} \in \mathcal{R}_+^{n\times c}, \mathbf{Y}^T\mathbf{Y} = \mathbf{I}_c, \mathbf{W} \in \mathcal{R}^{d\times c}, \end{aligned} \tag{5.1}$$

where $\nu, \lambda \in \mathcal{R}_+$ are parameters. Feature selection is performed through joint adaptive loss and l_2/l_0-norm minimization. The sparsity-inducing property of l_2/l_0-norm pushes the feature selection matrix $\mathbf{W}$ to be sparse in rows. More specifically, $\mathbf{w}^j$ shrinks to zero if the j-th feature is less correlated to the pseudo labels $\mathbf{Y}$. We can thus filter out the features corresponding to zero rows of $\mathbf{W}$ when performing feature selection. $\mathbf{L}$ is a Laplacian matrix which incorporates the neighborhood information on the data graph such that geometrically similar examples belong to similar pseudo cluster labels. In manifold learning, graph Laplacian is defined by $\mathbf{L} = \mathbf{D} - \mathbf{A}$, where A_{ij} is the edge weight between $\mathbf{x}_i$ and $\mathbf{x}_j$ in the sparse adjacency matrix on the neighborhood graph (e.g., one can use Gaussian kernel or K-nearest neighbors) and $\mathbf{D}$ is a diagonal matrix with its i-th diagonal entry being $D_{ii} = \sum_{j=1}^{n} A_{ij}$. Normalized graph Laplacian is defined by $\tilde{\mathbf{L}} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$. Graph Laplacian has been extensively used in semi-supervised learning [79, 80, 81] and unsupervised learning [82]. $\mathbf{L}$ can be traditional Laplacian matrix or local learning regularization matrix [46, 83]. For simplicity, we use normalized Laplacian matrix previously defined.

5.4 Optimization Algorithm

We adopt an alternating optimization strategy to solve AUFS.

5.4.1 Updating Weight Matrix W

Given fixed $\mathbf{Y}$, we update $\mathbf{W}$ by solving the subproblem

$$\min_{\mathbf{W}} \lambda \|\mathbf{X}\mathbf{W} - \mathbf{Y}\|_{\sigma} + \nu \|\mathbf{W}\|_{2,0}. \tag{5.2}$$

Let $f(\mathbf{W}) = \lambda \|\mathbf{X}\mathbf{W} - \mathbf{Y}\|_{\sigma}$, and $\phi(\mathbf{W}) = \|\mathbf{W}\|_{2,0}$. Assume that $f(\mathbf{W})$ is Lipschitz continuous, i.e.,

$$\|\nabla f(\mathbf{W}) - \nabla f(\mathbf{V})\|_F \leq L_f \|\mathbf{W} - \mathbf{V}\|_F \; \forall \, \mathbf{W}, \mathbf{V}.$$

By applying Taylor's theorem (define $\mathbf{V} = \mathbf{W} - \mathbf{W}^t$), we have

$$\begin{aligned} f(\mathbf{W}) &= f\left(\mathbf{W}^t\right) + \langle \mathbf{V}, \nabla f\left(\mathbf{W}^t\right)\rangle + \int_0^1 \langle \nabla f\left(\mathbf{W}^t + t\mathbf{V}\right) - \nabla f\left(\mathbf{W}^t\right), \mathbf{V}\rangle dt \\ &\leq f\left(\mathbf{W}^t\right) + \langle \mathbf{V}, \nabla f\left(\mathbf{W}^t\right)\rangle + \int_0^1 \left\|\nabla f\left(\mathbf{W}^t + t\mathbf{V}\right) - \nabla f\left(\mathbf{W}^t\right)\right\|_F \|\mathbf{V}\|_F dt \\ &\leq f\left(\mathbf{W}^t\right) + \langle \mathbf{V}, \nabla f\left(\mathbf{W}^t\right)\rangle + \int_0^1 L_f t \, \|\mathbf{V}\|_F^2 \, dt \\ &= f\left(\mathbf{W}^t\right) + \langle \mathbf{V}, \nabla f\left(\mathbf{W}^t\right)\rangle + \frac{L_f}{2} \|\mathbf{V}\|_F^2 . \end{aligned}$$

Therefore, a quadratic upper bound can be obtained for f:

$$f\left(\mathbf{W}^{t+1}\right) \leq f\left(\mathbf{W}^t\right) + \langle \mathbf{W}^{t+1} - \mathbf{W}^t, \nabla f\left(\mathbf{W}^t\right)\rangle + \frac{L_f}{2} \left\|\mathbf{W}^{t+1} - \mathbf{W}^t\right\|_F^2, \; \forall \, \mathbf{W}^{t+1}. \tag{5.3}$$

Since adaptive loss is differentiable, the gradient $\nabla f\left(\mathbf{W}^t\right)$ can be computed by

$$\nabla f\left(\mathbf{W}^t\right) = \lambda \mathbf{X}^T \mathbf{D}^t \left(\mathbf{X}\mathbf{W}^t - \mathbf{Y}\right), \tag{5.4}$$

where $\mathbf{D}^t$ is a diagonal matrix with

$$D_{ii}^t = \frac{(1+\sigma)\left(\left\|\mathbf{x}^i\mathbf{W}^t - \mathbf{y}^i\right\|_2 + 2\sigma\right)}{\left(\left\|\mathbf{x}^i\mathbf{W}^t - \mathbf{y}^i\right\|_2 + \sigma\right)^2}.$$

Defining function $g(\mathbf{W})$ by

$$f(\mathbf{W}^t) + \langle \mathbf{W} - \mathbf{W}^t, \nabla f(\mathbf{W}^t) \rangle + \frac{L_f}{2} \|\mathbf{W} - \mathbf{W}^t\|_F^2 + \nu\phi(\mathbf{W}),$$

we can show that if

$$\mathbf{W}^{t+1} = \arg\min g(\mathbf{W}) = \text{prox}_{\frac{\nu}{L}\phi}\left(\mathbf{W}^t - \frac{1}{L_f}\nabla f(\mathbf{W}^t)\right),$$

$$\begin{aligned} \text{then } f(\mathbf{W}^{t+1}) + \nu\phi(\mathbf{W}^{t+1}) &\leq g(\mathbf{W}^{t+1}) \\ &\leq g(\mathbf{W}^t) \\ &= f(\mathbf{W}^t) + \nu\phi(\mathbf{W}^t). \end{aligned}$$

The first inequality is obtained by adding $\nu\phi(\mathbf{W}^{t+1})$ on both sides of Eq. (5.3). The second inequality is from the definition of $\mathbf{W}^{t+1}$. Since L_f is unknown beforehand, we can initialize it by L^0, and increase it until the above inequality holds.

The proximal operator [84] for the function $\frac{\nu}{L}\|\cdot\|_{2,0}$ has a closed form solution:

$$\text{prox}_{\frac{\nu}{L}\|\cdot\|_{2,0}}(\mathbf{a}^i) = \begin{cases} \mathbf{0}, & \text{if } \|\mathbf{a}^i\|_2^2 \leq \frac{2\nu}{L} \\ \mathbf{a}^i, & \text{otherwise} \end{cases}.$$

Note that due to generality of proximal mapping, many other sparsity-inducing norms such as $l_{2,1}$-norm and $l_{\infty,1}$-norm can be used without changing the main algorithm framework, in which sense AUFS is a general framework where multiple sparsity-inducing norms can be used, whereas the optimization algorithms by UDFS, NDFS, and RUFS can only deal with $l_{2,1}$-norm.

The procedure to update $\mathbf{W}$ is listed in Algorithm 5.

Algorithm 5 Algorithm of Updating $\mathbf{W}$.

Input: $\mathbf{W}^0$, $\gamma = 1.5$, $L^0 = 1$, $\varepsilon = 10^{-2}$
$\phi(\cdot) \leftarrow \|\cdot\|_{2,0}$
$L \leftarrow L^0$
$t \leftarrow 0$
repeat
  repeat
    $L \leftarrow \gamma L$
    $\mathbf{W}^{t+1} \leftarrow \text{prox}_{\frac{\nu}{L}\phi}\left(\mathbf{W}^t - \frac{1}{L}\nabla f\left(\mathbf{W}^t\right)\right)$
  until $f\left(\mathbf{W}^{t+1}\right) + \nu\phi\left(\mathbf{W}^{t+1}\right) \leq f\left(\mathbf{W}^t\right) + \nu\phi\left(\mathbf{W}^t\right)$
  $t \leftarrow t + 1$
until $\left\|\mathbf{W}^t - \mathbf{W}^{t-1}\right\|_F \leq \frac{\varepsilon}{L}$
Output: $\mathbf{W}_t$

5.4.2 Updating Label Indicator Matrix Y

By Theorem 1 in [78], solving the following subproblem on the indicator matrix $\mathbf{Y}$ with fixed $\mathbf{W}$ will monotonically decrease the objective of the problem (5.1):

$$\begin{aligned} \min \quad & \text{Tr}\left[\mathbf{Y}^T\mathbf{L}\mathbf{Y}\right] + \lambda\text{Tr}\left[(\mathbf{X}\mathbf{W} - \mathbf{Y})^T\mathbf{D}\left(\mathbf{X}\mathbf{W} - \mathbf{Y}\right)\right] \\ \text{s.t.} \quad & \mathbf{Y}^T\mathbf{Y} = \mathbf{I}, \mathbf{Y} \geq 0, \end{aligned} \tag{5.5}$$

where $\mathbf{D}$ is a diagonal matrix with

$$D_{ii} = \frac{(1+\sigma)\left(\left\|\mathbf{x}^i\mathbf{W} - \mathbf{y}_t^i\right\|_2 + 2\sigma\right)}{2\left(\left\|\mathbf{x}^i\mathbf{W} - \mathbf{y}_t^i\right\|_2 + \sigma\right)^2}.$$

Denote the objective function in problem (5.5) by $J(\mathbf{Y})$, the Lagrange function is given by

$$\mathscr{L}(\mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Sigma}) = J(\mathbf{Y}) - \text{Tr}\left[\mathbf{\Lambda}\left(\mathbf{Y}^T\mathbf{Y} - \mathbf{I}\right)\right] - \text{Tr}\left[\mathbf{\Sigma}^T\mathbf{Y}\right],$$

where the symmetric matrix variable $\mathbf{\Lambda}$ is the Lagrange multiplier w.r.t. the orthogonal constraint, and the nonnegative matrix variable $\mathbf{\Sigma}$ is the Lagrange multiplier corresponding to the nonnegative

constraint. The Karush-Kuhn-Tucker conditions for problem (5.5) are given by

$$\begin{cases} \nabla J(\mathbf{Y}) - 2\mathbf{Y}\mathbf{\Lambda} - \mathbf{\Sigma} = \mathbf{0} \\ \mathbf{Y}^T\mathbf{Y} = \mathbf{I} \\ \mathbf{\Sigma} \odot \mathbf{Y} = \mathbf{0};\ \mathbf{\Sigma} \geq \mathbf{0};\ \mathbf{Y} \geq \mathbf{0} \end{cases}.$$

Using the KKT complementary slackness condition we have

$$\left(\frac{\partial J}{\partial \mathbf{Y}} - 2\mathbf{Y}\mathbf{\Lambda}\right)_{ik} Y_{ik} = 0,$$

which gives the diagonal entries of $\mathbf{\Lambda}$

$$\mathbf{\Lambda}_{kk} = \frac{1}{2}\left(\mathbf{Y}^T \frac{\partial J}{\partial \mathbf{Y}}\right)_{kk}.$$

For the off-diagonal entries, since the updated $\mathbf{Y}$ is guaranteed to be nonnegative, we can ignore $\mathbf{\Sigma}$, we thus have $\frac{\partial J}{\partial \mathbf{Y}} - 2\mathbf{Y}\mathbf{\Lambda} = \mathbf{0}$, giving $\mathbf{\Lambda} = \frac{1}{2}\mathbf{Y}^T\frac{\partial J}{\partial \mathbf{Y}}$. By substituting the gradient, we have

$$\mathbf{\Lambda} = \mathbf{Y}^T\mathbf{L}\mathbf{Y} + \lambda\mathbf{Y}^T\mathbf{D}\mathbf{Y} - \lambda\mathbf{Y}^T\mathbf{D}\mathbf{X}\mathbf{W}.$$

Since $\mathbf{Y}$ has a nonnegative constraint, we need to first decompose $\mathbf{W} = \mathbf{W}^+ - \mathbf{W}^-$, $\mathbf{L} = \mathbf{L}^+ - \mathbf{L}^-$, and $\mathbf{\Lambda} = \mathbf{\Lambda}^+ - \mathbf{\Lambda}^-$, where

$$\begin{aligned} \mathbf{\Lambda}^+ &= \mathbf{Y}^T\mathbf{L}^+\mathbf{Y} + \lambda\mathbf{Y}^T\mathbf{D}\mathbf{Y} + \lambda\mathbf{Y}^T\mathbf{D}\mathbf{X}\mathbf{W}^- \\ \mathbf{\Lambda}^- &= \mathbf{Y}^T\mathbf{L}^-\mathbf{Y} + \lambda\mathbf{Y}^T\mathbf{D}\mathbf{X}\mathbf{W}^+. \end{aligned}$$

Now concentrating on $\mathbf{Y}$ gives

$$\frac{1}{2}\frac{\partial}{\partial \mathbf{Y}}\left[J(\mathbf{Y}) - \mathrm{Tr}\mathbf{\Lambda}\left(\mathbf{Y}^T\mathbf{Y}\right)\right] = \mathbf{L}^+\mathbf{Y} + \lambda\mathbf{D}\mathbf{Y} + \lambda\mathbf{D}\mathbf{X}\mathbf{W}^- + \mathbf{Y}\mathbf{\Lambda}^- - \mathbf{L}^-\mathbf{Y} - \lambda\mathbf{D}\mathbf{X}\mathbf{W}^+ - \mathbf{Y}\mathbf{\Lambda}^+, \tag{5.6}$$

We thus obtain the following update formula for $\mathbf{Y}$ by applying the auxiliary function approach in [85]

$$Y_{ik} \leftarrow Y_{ik} \frac{[\mathbf{L}^{-}\mathbf{Y} + \lambda \mathbf{D}\mathbf{X}\mathbf{W}^{+} + \mathbf{Y}\mathbf{\Lambda}^{+}]_{ik}}{[\mathbf{L}^{+}\mathbf{Y} + \lambda \mathbf{D}\mathbf{X}\mathbf{W}^{-} + \lambda \mathbf{D}\mathbf{Y} + \mathbf{Y}\mathbf{\Lambda}^{-}]_{ik}}. \tag{5.7}$$

followed by column-wise normalization. We can see that Y_{ik} decreases when the corresponding element of the gradient in Eq. (5.6) is positive, and increases otherwise. Thus the update direction is consistent to the one in gradient descent. Our extensive experiments show that the iterative algorithm presented here always converges and monotonically increases the objective function in each iteration. When converges, we have

$$\left(\nabla J\left(\mathbf{Y}\right) - 2\mathbf{Y}\mathbf{\Lambda}\right) \odot \mathbf{Y} = \mathbf{0},$$

which is exactly the KKT complementary slackness condition.

Algorithm 6 AUFS Algorithm.

Input: $\mathbf{X} \in \mathcal{R}^{n \times d}, \mathbf{L}, \sigma, \lambda, \nu, \tau$ and p

Initialize $\mathbf{Y}_0$ (e.g., by K-means)
Initialize $\mathbf{W}_0$
$k \leftarrow 0$
repeat
Fixing $\mathbf{Y}_k$, compute $\mathbf{W}_{k+1}$ from
Algorithm 5 given $\mathbf{Y}_k, \mathbf{W}_k, \lambda$, and ν
Fixing $\mathbf{W}_{k+1}$, compute $\mathbf{Y}_{k+1}$ by Eg. (5.7)
$k \leftarrow k + 1$
until $\left\| \left(T_{\mathcal{R}_{+}^{n \times c}} \nabla \mathbf{Y}_k, \nabla \mathbf{W}_k \right) \right\| \leq \tau$

Output: Sort all d features according to $\|\mathbf{w}_k^j\|_2$ in descending order and select the top p ranked features.

For stopping criterion, define projection operator by

$$[T_{\Omega}\mathbf{M}]_{ij} = \begin{cases} M_{ij} & \text{if} \quad X_{ij} > 0 \\ \min\{M_{ij}, 0\} & \text{if} \quad X_{ij} = 0 \end{cases}. \tag{5.8}$$

Given a tolerance τ, an approximate solution to problem (5.1) is any matrix pair $(\mathbf{Y}, \mathbf{W})$ such that

$$\left\| \left(T_{\mathcal{R}_{+}^{n \times c}} \nabla \mathbf{Y}, \nabla \mathbf{W} \right) \right\| \leq \tau. \tag{5.9}$$

Table 5.1: Complexity of AUFS and the state-of-the-arts, where n is sample size, d is feature size and c is number of classes, k_{nn} is the number of nearest neighbors when constructing the sparse adjacency matrix for computing the Laplacian matrix.

Method	Computation	Memory
UDFS	$O(d^3) + O(cnk_{nn})$	$O(d^2) + O(nk_{nn}) + O(cn)$
NDFS	$O(d^3) + O(cnk_{nn})$	$O(d^2) + O(nk_{nn}) + O(cn)$
RUFS	$O(cnk_{nn} + ndc + nc^2)$	$O(nk_{nn} + nd + nc + dc)$
AUFS	$O(cnk_{nn} + ndc + nc^2)$	$O(nd + dc + c^2 + nc + nk_{nn})$

Finally, the iterative algorithm for AUFS is listed in Algorithm 6.

5.4.3 Complexity Analysis

For computation complexity, updating $\mathbf{W}$ for each iteration requires computing partial gradient in Eq. (5.4) and evaluation of objective function in Eq. (5.2), and they both take $O(ndc)$. Updating $\mathbf{Y}$ for each iteration requires matrix multiplication, which takes $O(cnk_{nn} + ndc + nc^2)$, where k_{nn} is the number of nearest neighbors when constructing the sparse adjacency matrix for computing the Laplacian matrix. Therefore, denoting t the number of outer iterations of Algorithm 6, the total computation complexity is $O(cnk_{nn} + ndc + nc^2)$. For memory cost, updating $\mathbf{W}$ requires $O(nd + dc + nc)$, and updating $\mathbf{Y}$ requires $O(nd + dc + c^2 + nc + nk_{nn})$, thus the total memory cost is $O(nd + dc + c^2 + nc + nk_{nn})$. For comparison, both UDFS and NDFS require $O(d^3) + O(cnk_{nn})$ computation complexity and $O(d^2) + O(nk_{nn}) + O(cn)$ memory cost. For RUFS, the computation complexity is $O(cnk_{nn} + ndc + nc^2)$ and its memory cost is $O(nk_{nn} + nd + nc + dc)$. Table 5.1 lists the complexity of AUFS and the state-of-the-arts.

Since the computational complexity and memory cost of AUFS is linear to the feature size d and the data size n, the proposed method can be run on big data. The only restriction is the requirement that data and intermediate matrices should be stored in memory since it is a sequential and iterative algorithm. In this case, one can use Apache Spark to process large scale datasets as it supports cyclic data flow and in-memory computing.

Table 5.2: Dataset Description.

Dataset	# of Samples	# of Features	# of Classes
BinaryAlphadigits	1404	320	36
COIL20	1440	1024	20
JAFFE	213	676	10
Pointing04	2790	1120	15
UMIST	575	644	20
USPS	11000	256	10
WebKB4	4199	1000	4

5.5 Experiments

As in previous unsupervised feature selection work [5, 6, 75], we evaluate the performance of AUFS for feature selection on clustering.

5.5.1 Datasets

The evaluation is performed on 7 benchmark real world datasets including Binary Alphadigits[1], COIL20[2], JAFFE[3], Pointing04[4] UMIST[5], USPS[6], and WebKB4 [46]. Detailed information is summarized in Table 5.2.

5.5.2 Compared Methods

We compare AUFS with the following unsupervised feature selection algorithms.

1. **Baseline**: All original features are adopted.
2. **MCFS**: Mutli-Cluster Feature Selection [4] where features are selected using spectral regression with l_1-norm regularization.
3. **UDFS**: Unsupervised Discriminative Feature Selection [5] which exploits local discriminative information and feature correlations simultaneously and considers the manifold structure as well.

[1] http://www.cs.nyu.edu/∼roweis/data.html
[2] http://www.cad.zju.edu.cn/home/dengcai/Data/MLData.html
[3] http://www.kasrl.org/jaffe.html
[4] http://www-prima.inrialpes.fr/Pointing04/data-face.html
[5] http://www.sheffield.ac.uk/eee/research/iel/research/face
[6] http://www.cad.zju.edu.cn/home/dengcai/Data/MLData.html

Table 5.3: Clustering Results (ACC% $\pm$ std). $*$ indicates statistical significance at the 5% level.

Dataset	All Features	MCFS	UDFS	NDFS	RUFS	AUFS
BinaryAlphadigits	41.1 ± 1.7	42.2 ± 2.0	42.6 ± 2.5	42.6 ± 2.0	43.1 ± 1.7	$\mathbf{43.7 \pm 1.7}$
COIL20	59.5 ± 4.4	57.8 ± 3.4	59.1 ± 3.4	60.8 ± 3.6	60.9 ± 5.3	$\mathbf{66.6 \pm 2.6}^*$
JAFFE	74.2 ± 7.3	76.5 ± 10.9	76.0 ± 8.1	76.4 ± 10.8	76.9 ± 10.6	$\mathbf{79.3 \pm 10.4}$
Pointing04	45.7 ± 3.8	62.8 ± 3.7	56.8 ± 3.2	61.0 ± 2.9	64.9 ± 4.3	$\mathbf{66.6 \pm 2.6}$
UMIST	42.3 ± 2.3	46.3 ± 2.7	44.9 ± 2.6	48.9 ± 3.4	47.5 ± 2.9	$\mathbf{49.8 \pm 3.1}^*$
USPS	44.6 ± 2.6	47.9 ± 2.6	44.1 ± 3.4	45.8 ± 1.6	48.3 ± 1.7	$\mathbf{51.4 \pm 1.8}^*$
WebKB4	65.8 ± 7.7	64.1 ± 6.7	59.0 ± 10.1	70.4 ± 8.4	74.0 ± 3.3	$\mathbf{80.4 \pm 3.5}^*$

Table 5.4: Clustering Results (NMI% $\pm$ std). $*$ indicates statistical significance at the 5% level.

Dataset	All Features	MCFS	UDFS	NDFS	RUFS	AUFS
BinaryAlphadigits	57.9 ± 0.7	58.1 ± 1.1	59.0 ± 0.9	58.8 ± 1.1	59.3 ± 0.6	$\mathbf{59.7 \pm 1.1}$
COIL20	75.6 ± 1.8	73.8 ± 1.6	74.1 ± 2.4	76.5 ± 1.5	76.5 ± 2.1	$\mathbf{77.6 \pm 1.2}^*$
JAFFE	82.5 ± 3.4	84.2 ± 5.4	84.0 ± 3.7	83.6 ± 5.8	84.3 ± 5.1	$\mathbf{85.7 \pm 4.9}$
Pointing04	50.0 ± 2.4	72.8 ± 2.3	64.5 ± 2.0	71.3 ± 2.1	76.0 ± 1.6	$\mathbf{76.4 \pm 1.9}$
UMIST	64.1 ± 1.7	67.5 ± 1.8	65.3 ± 1.7	67.4 ± 1.7	68.4 ± 2.2	$\mathbf{69.8 \pm 1.8}^*$
USPS	44.8 ± 1.7	44.9 ± 1.8	43.0 ± 1.7	45.1 ± 1.2	46.0 ± 1.1	$\mathbf{47.5 \pm 1.8}^*$
WebKB4	45.7 ± 2.8	42.8 ± 3.0	33.5 ± 8.4	44.0 ± 10.4	49.8 ± 2.2	$\mathbf{58.7 \pm 1.8}^*$

4. **NDFS**: Nonnegative Discriminative Feature Selection [6] where features are selected by a joint framework of nonnegative spectral analysis and $l_{2,1}$-norm regularized regression.

5. **RUFS**: Robust Unsupervised Feature Selection [75] which selects features by joint local learning regularized robust NMF and joint $l_{2,1}$-norm minimization.

5.5.3 Experimental Settings

Following previous work, Accuracy (ACC) and Normalized Mutual Information (NMI) are used for evaluation [4].

As for experimental settings, for MCFS, UDFS, NDGS, RUFS, and AUFS we fix the neighborhood size $k = 5$ for all datasets. When computing the weight matrix for the data graph, following standard graph Laplacian construction, we use cosine kernel for text data and Gaussian kernel for other types of data. c is set to the number of clusters. For AUFS, we fix $\sigma = 1$ for the adaptive loss. Since label information is supposed unavailable,, we cannot use a validation set to tune the parameters, we thus search all parameters over the grid and report the best result it can achieve. Specifically, we tune the parameters for all methods by a "grid-search" strategy from $\{10^{-6}, 10^{-4}, \ldots, 10^{4}, 10^{6}\}$. The number of selected features are set as $\{50, 100, 150, \ldots 300\}$ for all datasets except USPS where $\{50, 80, 110, 140, 170, 200\}$ is used. Though such a strategy

could be biased, it is still a fair comparison because we did this for all the methods as long as there're paramters to tune. Best clustering results from the optimal parameters are reported for all methods. For evaluation, we use K-means on the selected features. Since K-means depends on initialization, following previous work, we repeat clustering 20 times with random initialization for each setup. The average results with standard deviation are reported.

5.5.4 Results and Discussion

From the experimental results shown in Table 5.3 and Table 5.4 we can draw a conclusion that feature selection is important and effective. Not only can the number of features be significantly reduced, making posterior processing more efficient, but clustering performance can also be greatly improved. We also see that AUFS achieves the best performance, which validates our analysis in the motivation. In fact, NDFS's objective function is similar to AUFS except that it uses joint squared Frobenius norm and $l_{2,1}$-norm to select features (both methods use non-negative spectral analysis to learn pseudo label). In this case, we have already empirically shown that the improvement originates from the new objective function.

We also study the sensitiveness of parameters. We only report the results on COIL20 dataset with on Figure 5.2 (similar results can be observed for other datasets). The experimental results show that AUFS is not very sensitive to λ with wide ranges when λ is not large, which is reasonable because larger λ weakens the effect of the graph Laplacian term thus hurts the quality of the learned pseudo labels. AUFS is also not very sensitive to ν with wide ranges when ν is not large, but large ν does badly hurt the performance for larger ν favors zero weight matrix. In practice, we suggest using a validation set with ground truth under an affordable cost to tune the parameters by e.g. grid search. Again, different users may label the data points differently, we can ask similar users to construct the validation set to tune parameters that work best for them.

We then study the convergence of AUFS. We report convergence curves for all datasets in Figure 5.3 and Figure 5.4. We can see that the proposed optimization algorithm converges quickly.

Finally, we compare the average running time of different methods on all 7 datasets in Table 5.5. We set maximum number of iterations to 50 and tolerance to 10^{-4} times the norm of the initial gradient, and an algorithm terminates whichever comes first. The computation is performed using

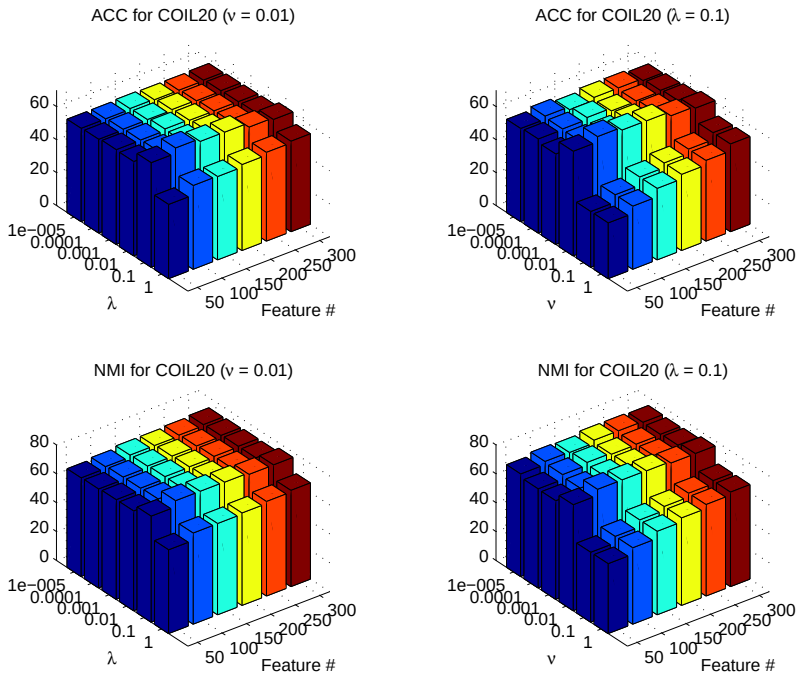


Figure 5.2: ACC and NMI of AUFS with different λ, ν and feature numbers on COIL20 dataset.

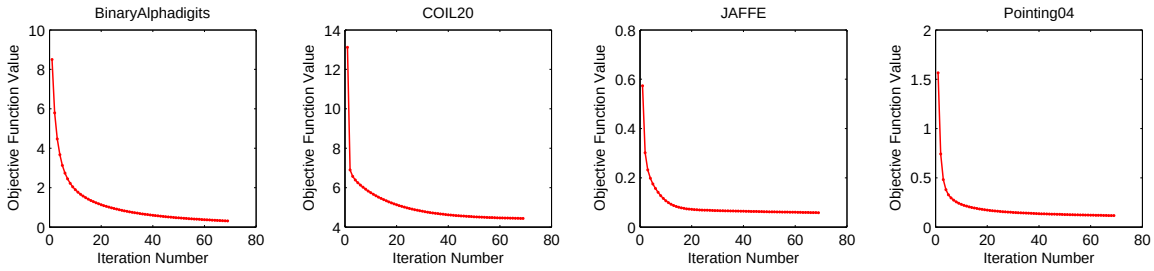


Figure 5.3: Convergence curves of AUFS on BinaryAlphadigits, COIL20, JAFFE, and Pointing04 datasets.

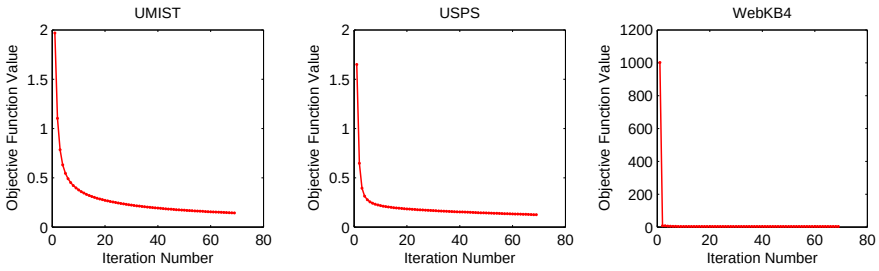


Figure 5.4: Convergence curves of AUFS on UMIST, USPS, and WebKB4 datasets.

Table 5.5: Average Running Time (seconds).

Dataset	UDFS	NDFS	RUFS	AUFS
BinaryAlphadigits	11.3	**10.5**	14.3	13.4
COIL20	35.9	31.9	27.2	**6.8**
JAFFE	2.7	**1.9**	4.8	2.7
Pointing04	174.3	84.1	232.0	**69.9**
UMIST	**5.9**	7.1	17.0	7.2
USPS	283.1	168.7	**36.0**	136.8
WebKB4	375.1	203.9	88.7	**30.4**

a 3.6GHz (2 x 4cores), 96GB memory with Linux OS. Table 5.5 shows that AUFS is competitive to the fastest algorithm on small datasets, and it outperforms all state-of-the-art methods on datasets with large sample size and large feature size, though RUFS is much faster than AUFS on USPS dataset which has a relatively small feature size (objective function value of RUFS doesn't decrease on USPS which results in a fake termination).

5.5.5 Limitation

Note that only small-scale public benchmark datasets are used in the experiments because researchers in the literature usually compare algorithms on these datasets. As is mentioned in the last chapter, one advantage is that it is more convenient for people to assess and compare different methods. Another advantage is that it doesn't require expensive distributive system for evaluation, which would allow more researchers to test the algorithms. However, since the experiments are conduct on small-scale datasets, the algorithms' performance rank might not be preserved on large scale datasets. But it can be qualitatively argued that the proposed method would still outperform the baseline methods because the distribution of outliers on large scale datasets would be similar if it is not exactly the same to these small-scale benchmark real world datasets. For example, the distribution of outliers of a large scale ORL dataset will be highly expected to be similar to the small scale ORL dataset used in this experiment. Thus a method which is able to achieve good balance between outliers and normal examples is potentially superior to the one without this property. For quantitative validation, experiments on large-scale datasets need to be done before conclusions could be made on large-scale datasets, which is a promising research direction for future work.

5.6 Summary

We propose a new unsupervised feature selection approach called AUFS, which jointly minimizes the adaptive loss and l_2/l_0-norm. We directly solve the nonnegative orthogonal constrained optimization problem so that more accurate and discriminative pseudo labels can be learned. We derive an effective and efficient iterative algorithm to make AUFS be applicable for large scale feature selection tasks. Extensive experimental results on different real world datasets validate the effectiveness and efficiency of the new method.

We have discussed unsupervised feature selection for single-view data in the previous two chapters. In the next two chapters, we will study unsupervised feature selection and topic discovery multi-view data.

Chapter 6

Unsupervised Feature Selection for Multi-View Clustering on Text-Image Web News Data

In the last two chapters, we studied unsupervised feature selection from single-view perspective; in this chapter, we look at the problem from multi-view perspective. Specifically, we study how to do feature selection on text-image web news data when labels are not available.

6.1 Introduction

In this chapter we extends the first and the second work in the sense that we want to propose a more effective unsupervised feature selection method for high dimensional multi-view data, and particularly we focus on text-image web news data [82]. Reading web news articles is an important part of people's daily life, especially in the current "big data" era that we are facing a large amount of information every day due to the advancement and development of information technology. One ideal way is to automatically group the web news per their content into multiple clusters, e.g., technology and health care, then one can choose to read the latest and the most representative news articles in a group of interest. This procedure can be done recursively so that one can explore the news in different resolution hierarchically. Clustering web news is also an effective way to organize, manage, and search news articles. Unlike traditional document clustering, images play an important role in web news articles as is evident from the fact that almost all news articles have one picture associated. How to effectively and efficiently group web news articles of multiple modality is challenging because different data types have different properties and different feature spaces and also because the dimensionality of feature spaces is usually very high. For example in text feature space, the vocabulary size can be over a million. Besides, there are a lot of unrelated and noisy features which often lead to low efficiency and poor performance.

Multi-view unsupervised feature selection is desirable to solve the problem mentioned above,

since it can select most discriminative features while considering the consensus from data of multiple views in an unsupervised fashion. Feature size can be extremely reduced and feature quality can be greatly enhanced. As a result, not only computation can be more efficient but clustering performance can also be greatly improved. However, not much work have been done to be able to solve this problem well, especially for multi-view clustering on web news data. State-of-the-art unsupervised feature selection methods [7, 8] for multi-view data use spectral clustering across different views to learn the most consistent pseudo class labels and simultaneously use the learned labels to do feature selection. More specifically, Adaptive Unsupervised Multi-view Feature Selection (AUMFS) [7] uses spectral clustering on a combined data similarity graph from different views to learn the labels that have most consensus across different views, and then use $l_{2,1}$-norm regularized robust sparse regression to learn one weight matrix for all the features of different views to best approximate the cluster labels. [8] presents a new unsupervised multi-view feature selection method called Multi-View Feature Selection (MVFS). MVFS also uses spectral clustering on the combined data similarity graph from different views to learn the labels, but learn one weight matrix for each view to best fit the learned pseudo class labels by joint squared Frobenius norm (fitting term) and $l_{2,1}$-norm (rowise sparsity-inducing). Both [7] and [8] share the disadvantage that they're sensitive to the combined data similarity graph, especially when there are quite a number of unrelated and noisy features in the feature space, and there is information loss during graph construction.

We propose to directly utilize raw features in the main view (e.g., text for text-image web news data) to learn pseudo cluster labels which should also have the most consensus with other views (e.g., image), and meanwhile the discriminative features in the feature selection process will win out to contribute more on label learning process, and in return the improved cluster labels will help to select more discriminative features for each view. Technically, we propose a new method called Multi-View Unsupervised Feature Selection (MVUFS) to do unsupervised feature selection for multi-view clustering, especially focused on analyzing text-image web news data. We propose to minimize the sum of regularized data matrix factorization error and data fitting error in a unified optimization setting. We use local learning regularized orthogonal nonnegative matrix factorization to learn pseudo cluster labels and simultaneously learn rowise sparse weight matrices for each view

by joint $l_{2,1}$-norm minimization guided by the learned pseudo cluster labels. The label learning process and feature selection process are mutually enhanced. For label learning, we factorize the data matrix in the main view (e.g. text) and ensure that the learned indicator matrix is as consistent as local learning predictors on other views (e.g. image). To objectively evaluate the new method, we build two text-image web news datasets from two major US news media web sites: CNN and FOXNews. Our extensive experiments show that MVUFS significantly outperforms state-of-the-art single-view and multi-view unsupervised feature selection methods.

6.2 Optimization Problem

MVUFS solves the following optimization problem:

$$\begin{aligned} \min \quad & \|\mathbf{X}_1 - \mathbf{G}\mathbf{F}\|_F^2 + \mathrm{Tr}\left[\mathbf{G}^T\mathbf{L}_2^{llr}\mathbf{G}\right] + \alpha\sum_{v=1}^{2}\|\mathbf{G} - \mathbf{X}_v\mathbf{W}_v\|_{2,1} + \beta\sum_{v=1}^{2}\|\mathbf{W}_v\|_{2,1} \\ \text{s.t.} \quad & \mathbf{G}^T\mathbf{G} = \mathbf{I}_c, \mathbf{G} \geq 0, \mathbf{F} \geq 0, \mathbf{W}_v \in \mathcal{R}^{d_v \times c} \end{aligned} \tag{6.1}$$

where α, β are nonnegative parameters and $\mathbf{L}_2^{llr}$ is the Laplacian matrix for local learning regularization on image view which can be computed by Eq. (4.1). To learn the most consistent pseudo labels across different views, we use orthogonal nonnegative matrix factorization on the text view regularized by local learning prediction error on the image view. $\mathbf{F}$ is the basis matrix with each row being a cluster center. The fitting term $\sum_{v=1}^{2}\|\mathbf{G} - \mathbf{X}_v\mathbf{W}_v\|_{2,1}$ will also push the pseudo labels to be close to the linear prediction by the feature weight matrices for each view, which gives the desirable mutual reinforcement between label learning and feature selection. Nonnegative and orthogonal constraints imposed on the cluster indicator matrix variable are desirable to give a single non-zero positive entry on each row of the label matrix. For feature selection, we adopt joint $l_{2,1}$-norm minimization [21] to learn rowise sparse weight matrices for each view. The sparsity-inducing property of l_2/l_1-norm pushes the feature selection matrix $\mathbf{W}_v$ to be sparse in rows. More specifically, $\mathbf{w}_v^j$ shrinks to zero if the j-th feature is less correlated to the pseudo labels $\mathbf{Y}$. We can thus filter out the features corresponding to zero rows of $\mathbf{W}_v$.

We apply alternating optimization to solve problem (6.1). To optimize $\mathbf{G}$ given $\mathbf{F}$, $\mathbf{W}_v$, $v = 1, 2$,

and $\mathbf{G}^t$ in the last iteration, we solve the following subproblem:

$$\begin{aligned} \min \quad & \|\mathbf{X}_1 - \mathbf{G}\mathbf{F}\|_F^2 + \mathrm{Tr}\left[\mathbf{G}^T\mathbf{L}_2^{llr}\mathbf{G}\right] + \alpha\sum_{v=1}^{2}\|\mathbf{D}_v\mathbf{G} - \mathbf{D}_v\mathbf{X}_v\mathbf{W}_v\|_F^2 \\ \text{s.t.} \quad & \mathbf{G}^T\mathbf{G} = \mathbf{I}_c, \mathbf{G} \geq 0, \end{aligned} \tag{6.2}$$

where $\mathbf{D}_v$ is a diagonal matrix: $D_{ii}^v = \frac{1}{2^{0.5}}\left\|\mathbf{g}_t^i - \mathbf{x}_v^i\mathbf{W}_v\right\|_2^{-0.5}$. It can be proved (due to space limit, we omit the proof) that if $\mathbf{G}^{t+1}$ is the solution of problem (6.2), $\mathbf{G}^{t+1}$ will monotonically decrease the objective function of problem (6.1). Denote the objective function in problem (6.2) by $J(\mathbf{G})$, the Lagrange function is given by $\mathscr{L}(\mathbf{G}, \mathbf{\Lambda}, \mathbf{\Sigma}) = J(\mathbf{G}) - \mathrm{Tr}\left[\mathbf{\Lambda}\left(\mathbf{G}^T\mathbf{G} - \mathbf{I}\right)\right] - \mathrm{Tr}\left[\mathbf{\Sigma}^T\mathbf{G}\right]$. The optimal $\mathbf{G}$ must satisfy the KKT condistions:

$$\begin{cases} \nabla J(\mathbf{G}) - 2\mathbf{G}\mathbf{\Lambda} - \mathbf{\Sigma} = \mathbf{0} \\ \mathbf{G}^T\mathbf{G} = \mathbf{I} \\ \mathbf{\Sigma} \odot \mathbf{G} = \mathbf{0};\ \mathbf{\Sigma} \geq \mathbf{0};\ \mathbf{G} \geq \mathbf{0} \end{cases}.$$

Since the updated $\mathbf{G}$ is guaranteed to be nonnegative, we can ignore $\mathbf{\Sigma}$, we thus have $\frac{\partial J}{\partial \mathbf{G}} - 2\mathbf{G}\mathbf{\Lambda} = \mathbf{0}$, giving $\mathbf{\Lambda} = \frac{1}{2}\mathbf{G}^T\frac{\partial J}{\partial \mathbf{G}}$. We first decompose $\mathbf{W}_v = \mathbf{W}_v^+ - \mathbf{W}_v^-$ and $\mathbf{\Lambda} = \mathbf{\Lambda}^+ - \mathbf{\Lambda}^-$, where

$$\begin{aligned} \mathbf{\Lambda}^+ &= \mathbf{G}^T\mathbf{G}\mathbf{F}\mathbf{F}^T + \mathbf{G}^T\mathbf{L}_2^{llr+}\mathbf{G} + \alpha\mathbf{G}^T\left(\sum_{v=1}^{2}\mathbf{D}_v^2\right)\mathbf{G} + \alpha\mathbf{G}^T\left(\sum_{v=1}^{2}\mathbf{D}_v^2\mathbf{X}_v\mathbf{W}_v^-\right) \\ \mathbf{\Lambda}^- &= \mathbf{G}^T\mathbf{X}_1\mathbf{F}^T + \mathbf{G}^T\mathbf{L}_2^{llr-}\mathbf{G} + \alpha\mathbf{G}^T\left(\sum_{v=1}^{2}\mathbf{D}_v^2\mathbf{X}_v\mathbf{W}_v^+\right). \end{aligned}$$

We then obtain the following update formula for $\mathbf{G}$ by applying the auxiliary function approach in [85]:

$$G_{ik} \leftarrow G_{ik}\frac{\left[\mathbf{X}_1\mathbf{F}^T + \mathbf{L}_2^-\mathbf{G} + \alpha\sum_{v=1}^{2}\mathbf{D}_v^2\mathbf{X}_v\mathbf{W}_v^+ + \mathbf{G}\mathbf{\Lambda}^+\right]_{ik}}{\left[\mathbf{G}\mathbf{F}\mathbf{F}^T + \mathbf{L}_2^+\mathbf{G} + \alpha\sum_{v=1}^{2}\mathbf{D}_v^2\mathbf{G} + \alpha\sum_{v=1}^{2}\mathbf{D}_v^2\mathbf{X}_v\mathbf{W}_v + \mathbf{G}\mathbf{\Lambda}^-\right]_{ik}}. \tag{6.3}$$

followed by column-wise normalization. When converges, we have $(\nabla J(\mathbf{G}) - 2\mathbf{G}\mathbf{\Lambda}) \odot \mathbf{G} = \mathbf{0}$, which is exactly the KKT complementary slackness condition.

To optimize $\mathbf{F}$, we solve the subproblem: $\min_{\mathbf{F}\geq 0}\|\mathbf{X}_1 - \mathbf{G}\mathbf{F}\|_F^2$. Since the objective function is quadratic, and $\mathbf{F}$'s columns are mutually independent, we can use blockwise coordinate descent

to update one row at a time in a cyclic order, and the objective function value is guaranteed to decrease. The updating formula for $\mathbf{F}$ is

$$\mathbf{F}_{i:} \leftarrow \max\left(\mathbf{0}, \mathbf{F}_{i:} - \frac{\left[\mathbf{G}^T\mathbf{G}\right]_{i:}\mathbf{F} - \left[\mathbf{G}^T\mathbf{X}_1\right]_{i:}}{\left[\mathbf{G}^T\mathbf{G}\right]_{ii}}\right). \tag{6.4}$$

To optimize $\mathbf{W}_v$, we need to solve the unconstrained problem $\min_{\mathbf{W}_v \in \mathcal{R}^{d_v \times c}} \alpha\|\mathbf{G} - \mathbf{X}_v\mathbf{W}_v\|_{2,1} + \beta\|\mathbf{W}_v\|_{2,1}$ for each view. There're several optimization strategies that can solve it. Here we adopt the simple algorithm given in [21].

Algorithm 7 MVUFS

Input: $\{\mathbf{X}_v, p_v\}_{v=1}^{2}, \mathbf{L}_2^{llr}, \alpha, \beta$
Output: p_v features for the v-th view, $v = 1, 2$

1: Initialize $\mathbf{G}^0$ s.t. $\mathbf{G}^{0\mathrm{T}}\mathbf{G}^0 = \mathbf{I}$ (e.g., by K-means) and $\mathbf{F}^0 = \mathbf{G}^{0T}\mathbf{X}_1$, $t \leftarrow 0$
2: **while** Not convergent **do**
3: Given $\mathbf{G}^t$ and $\mathbf{F}^t$, compute $\mathbf{W}_v^{t+1}$ as in [21]
4: Given $\mathbf{W}_v^{t+1}$ and $\mathbf{F}^t$, compute $\mathbf{G}^{t+1}$ by Eq. (6.3)
5: Given $\mathbf{W}_v^{t+1}$ and $\mathbf{G}^{t+1}$, compute $\mathbf{F}^{t+1}$ by Eq. (6.4)
6: $t \leftarrow t + 1$
7: **end while**
8: **for** $v = 1$ to 2 **do**
9: Sort all d_v features according to $\|\mathbf{w}_v^i\|_2$ in descending order and select the top p_v ranked features for the v-th view.
10: **end for**

6.2.1 Complexity Analysis

Updating $\mathbf{F}$ has $O(nc^2) + O(dc^2) + O(cdn)$ computational complexity and $O(nd + dc + nc)$ memory cost. Updating $\mathbf{G}$ has $O(cnk_{nn} + ndc + nc^2)$ computational complexity and $O(nd + dc + nc)$ memory cost. Updating $\mathbf{W}_v$ by joint $l_{2,1}$-norm minimization [21] has $O(cn^2)+O(cdn)$ computational complexity and $O(n^2+nd+dc+nc)$ memory cost resulted from solving a linear equation to obtain an $n \times c$ matrix variable. The bottleneck is the quadratic complexity for updating $\mathbf{W}_v$ when running on large scale datasets. The remedy can be devising a limited-memory BFGS based iterative algorithm to solve the joint $l_{2,1}$-norm minimization problem which would result in an algorithm of linear computational complexity and memory cost. Besides the quadratic issue, another restriction is the requirement that data and intermediate matrices should be stored in memory. In this case, one can use Apache Spark to process large scale datasets as it supports cyclic data flow and in-

Table 6.1: Dataset Description.

Dataset	# Instances	# Words	# IMG-features	# Classes
CNN	2107	7989	996	7
FOX	1523	5477	996	4

memory computing.

6.3 Experiments

6.3.1 Datasets

To the best of our knowledge, there's no public text-image web news datasets released when this work is being done, so we crawled CNN and FOXNews web news from Jan. 1st, 2014 to Apr. 4th, 2014. The category information contained in the RSS feeds for each news article can be viewed as reliable ground truth. Titles, abstracts, and text body contents are extracted as the text view data, and the image associated with the article is stored as the image view data. Since the vocabulary has a very long tail word distribution, We filtered out those words that occur less than or equal to 5 times. All text content is stemmed by portStemmer [86], and we use l_2-normalized TFIDF as text. For image features, we use 7 groups of color features: Color features include RGB dominant color, HSV dominant color, RGB color moment, HSV color moment, RGB color histogram, HSV color histogram, color coherence vector [87], and 5 textural features: four Tamura textural features [88] (coarseness, contrast, directionality, line-likeness) and Gabor transform [89, 90]. Statistics of CNN and FOX datasets are shown in Table 6.1.

Please be noted that there are only few news articles updated in the RSS feeds, and some web pages contains only videos, while some other pages provide no videos or images at all, which limits the scale of the datasets used in the experiments. Nevertheless, the conclusions would be expected to hold for large scale data for which we will give an explanation in the end of this section.

6.3.2 Settings

Two widely used evaluation metrics for measuring clustering performance: accuracy (ACC) and Normalized Mutual Information (NMI) are used. We compare MVUFS with KMeans on text with all features (KM-TXT), KMeans on image with all features (KM-IMG), state-of-the-art single view

Table 6.2: Clustering Results (ACC% $\pm$ std), $*$ means statistical significance at 5% level.

Dataset	KM-TXT	KM-IMG	NDFS	RUFS	MVSKM	AUMFS	MVFS	MVUFS
CNN	50.1 ± 7.2	23.2 ± 1.0	31.6 ± 6.1	31.3 ± 5.3	32.0 ± 2.8	54.2 ± 4.6	50.2 ± 4.8	$\mathbf{57.9 \pm 4.9}^*$
FOX	76.2 ± 7.7	43.0 ± 0.3	56.6 ± 9.3	61.2 ± 8.3	73.3 ± 2.1	83.7 ± 1.3	84.7 ± 0.6	$\mathbf{87.9 \pm 1.0}^*$

Table 6.3: Clustering Results (NMI% $\pm$ std), $*$ means statistical significance at 5% level.

Dataset	KM-TXT	KM-IMG	NDFS	RUFS	MVSKM	AUMFS	MVFS	MVUFS
CNN	42.0 ± 4.3	3.7 ± 0.1	21.1 ± 5.5	22.8 ± 4.9	16.6 ± 1.1	36.4 ± 3.2	30.8 ± 2.5	$\mathbf{44.1 \pm 2.4}^*$
FOX	67.3 ± 6.1	7.6 ± 0.3	37.3 ± 8.5	42.6 ± 12.5	50.0 ± 1.8	64.4 ± 0.9	66.5 ± 0.6	$\mathbf{72.1 \pm 0.5}^*$

unsupervised feature selection methods: NDFS [6] - Joint nonnegative spectral analysis and $l_{2,1}$-norm regularized regression and RUFS [75] - joint local learning regularized robust NMF and robust $l_{2,1}$-norm regression; multi-view spherical KMeans with all features (MVSKM) [91],
state-of-the-art multi-view unsupervised feature selection: AUMFS [7] - spectral clustering and $l_{2,1}$-norm regularized robust sparse regression and MVFS [8] - spectral clustering and $l_{2,1}$-norm regression. For single-view unsupervised feature selection methods, KMeans is used to calculate the clustering performance. For multi-view unsupervised feature selection methods, multi-view spherical KMeans [91] is used for multi-view clustering. We set the neighborhood size to be 5. We use cosine similarity to build text graph and Gaussian kernel for image graph. All feature selection methods have two parameters: α for regression, and β for sparsity control. We do grid search for α in $\left\{10^{-2}, 10^{-1}, \ldots, 10^{2}\right\}$, and β in $\alpha \times \left\{10^{-2}, 10^{-1}, \ldots, 10^{2}\right\}$. We vary the number of selected text features as $\{100, 300, 500, 700, 900\}$. The number of selected image features is half of selected text features. Since K-means depends on initialization, we repeat clustering 10 times with random initialization.

6.3.3 Results

We need to answer several questions. First, is multi-view clustering always better than single view clustering? From Table 6.2, Table 6.3, and Figure 6.1, we can see that the answer is no. It depends on the feature quality of different views. Here the color and texture features we used for image view is not tightly tied with clustering measures, which does severely hurt the performance of multi-view clustering (MVSKM behaves much worse than KM-TXT). Fortunately, if discriminative features are selected by using multi-view feature selection methods, the multi-view clustering performance

may be significantly improved and can be better than single-view performance. For example, MVUFS significantly outperforms all single-view methods. Second, is multi-view feature selection better than single-view feature selection? We see that AUMFS, MVFS, and MVUFS outperform standard single view features election methods such as NDFS and RUFS, which indicates that different views can mutually bootstrap each other. It's interesting to see that both NDFS and RUFS even behave worse than without doing feature selection. At last, it turns out that MVUFS outperforms both single-view clustering and feature selection methods and multi-view clustering and feature selection methods. Since the major difference between MVUFS and AUMFS, MVFS is label learning, we conclude that directly learning labels from raw features from one view while ensuring the most consensus with other views could select a more discriminative feature set for all views, and spectral clustering relies on the combined similarity graphs of all views which may result in loss of discriminative information and could undermine the performance.

6.3.4 Parameter Analysis

We plot ACC versus different α, β, and number of selected features on FOXNews for MVUFS in Figure 6.2 (similar figures for NMI and on CNN dataset) due to space limit. We see that an appropriate combination of these parameters is crucial. However, it is unknown to us theoretically how to choose the best parameter setting. It may depends on datasets and measures. In practice, like many other methods, one can build a validation set in a mild scale to tune parameters by e.g., grid search. Also as is discussed in previous chapters, the users may have different perspectives for clustering or classification, it is important to construct a validation set for consistent and similar users so that the tuned parameters could result in an optimal performance in their perspectives.

6.3.5 Limitation

Please be noted that one limitation of the experiment design is that the scale of the crawled dataset is still small thus the algorithms' performance rank might not be preserved on large scale datasets. But since the proposed method utilizes the detailed raw text features for label learning, this advantage would still take effect for large scale web new datasets, therefore the proposed method would still be superior to the baseline methods for large scale web new data. Of course

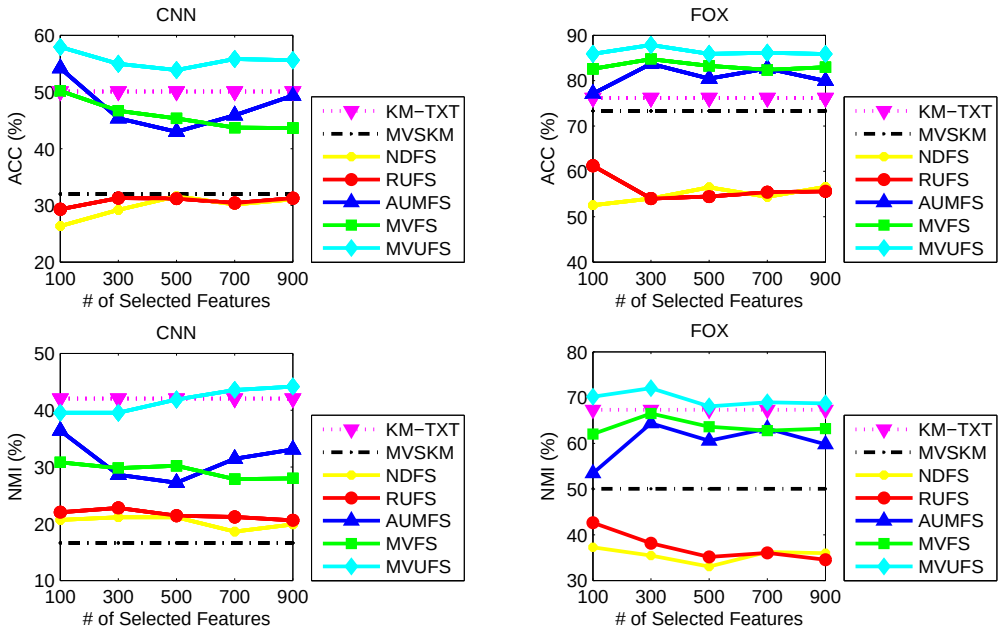


Figure 6.1: ACC and NMI with varying number of selected features.

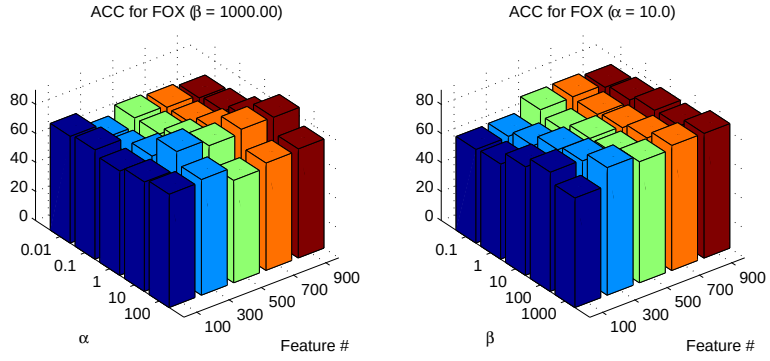


Figure 6.2: ACC v.s. different α, β, and number of selected features on FOX dataset for MVUFS.

for quantitative validation, experiments on large-scale datasets need to be done before conclusions could be made on large-scale datasets.

6.4 Summary

In this chapter, we propose a new unsupervised feature selection methods for multi-view clustering: MVUFS where local learning regularized orthogonal nonnegative matrix factorization is performed to learn pseudo class labels on raw features. We built two web news text-image datasets from CNN and FOXNews, and systematically evaluate MVUFS with state-of-the-art single-view and multi-view unsupervised feature selection methods. Experimental results validate the effectiveness of the proposed method.

In the next chapter we will present our work on topic discovery on multiple-view data.

Chapter 7

Text-Image Topic Discovery for Web News Data

In previous chapters, we discussed the problem of feature selection for unsupervised learning from both single-view and multi-view. In this chapter, we study unsupervised feature composition. As is previously mentioned, feature composition or topic discovery on single-view data has already been studied very well, and many famous topic models like PLSA and LDA are proposed in the literature. In contrast to single-view topic models, there is more space for topic discovery on multi-view data. In this chapter, we study unsupervised topic discovery on multi-view data, specifically on web news data.

7.1 Introduction

In practice, big data can be of multiple views. For example, web news articles contain not only text content but also have images associated. In this chapter we study how to systematically mine topics from high dimensional text-image web news data. Note that although the proposed formulation bases on web news data, the idea can be naturally extended to general multi-view data.

Exploring web news more efficiently and understanding them more effectively is important for absorbing information in our daily life. However, there're more and more web news articles but we have less and less time to read them. One ideal way is to automatically group the web news per their content or topics, then a user can choose a topic to read. This procedure can be done recursively so that a user could explore the news with least time. Images play an important role in news as is evident from the fact that almost all news articles have one picture associated. Thus to effectively organize news, it is important to consider both text and images. However, traditional topic modeling techniques such as LSI [31], PLSA [1], and LDA [2] are not powerful enough to handle heterogeneous data because they consider only the text content. Since modern web news

are usually composed of multiple data types such as text, image, and video, effective topic mining methods that can discover joint text-image topics and organize these multiple typed news data are urgently needed. The multiple typed topic discovery task is substantially different from topic modeling for a single text corpus because image is not a sequence of logical semantic units and image content is more difficult to numerically define and compute. Mining topics from heterogeneous data requires careful and insightful utilization of properties of every data types, which is not a trivial task.

Multi-view learning, which aims to learn better models to cluster data in multiple views, is a machine learning research area that can be applied for our problem, but state-of-the-art multi-view learning methods cannot do this task very well. Co-trained multi-view spectral clustering [39] iteratively uses the spectral embedding from one view to constrain the similarity graph used for the other view. However, this approach heavily relies on similarity graph for each view and completely ignores the detailed information, which may badly hurt the clustering performance due to loss of discriminative information. Also it's not straight forward to generate multi-view topic representation via this approach. [40] proposes to generalize K-means for multi-view data clustering. However, its performance tends to be dominated by the worst domain since the algorithm will assign large weight to the domain with the largest approximation error as will be demonstrated in the experiment later. There're also some heterogeneous data co-clustering work [41][42], however they require some supervision information. For example, [41] require user specified must-link and cannot-link constraint in the central type, and [42] require user preference before clustering. Besides, although heterogeneous co-clustering methods appear to be able to tackle our problem, they do not aim to explicitly learn representative and interpretable multi-view topics from heterogenous web news data. The major goal of this work is to provide an effective multi-view learning approach to discover text-image topics from web news data without any supervision.

7.2 Problem Formulation

In this section, we introduce the novel problem of joint text-image topic mining. We first formally define the new concept "text-image topic" and present a general optimization framework based on

regularized non-negative matrix factorization to sovle the problem.

Definition 7.2.1 (Text-image Topic)**.** *A text-image topic T is a bundle $\{V_1, V_2\}$, where V_1 is a weighted term vector or a set of weighted terms, V_2 is a set of selected images. Note that the concept of a text-image topic can be generalized as a multimedia topic which is a bundle $\{V_1, V_2, \ldots, V_M\}$ where V_1 is a weighted term vector or a set of weighted terms, V_2 is a set of selected images, V_3 is a set of selected videos, and V_M is the set of data description from the M-th media type.*

Definition 7.2.2 (Text-image Document)**.** *A text-image document D is a general form of text document which contains 2 mutually associated "subdocuments" corresponding to text and image media types. Formally, $D = \left(d^1, d^2\right)$, where d^1 is text, d^2 is images. The joint text-image document can be further generalized to multimedia documents, which contain M mutually associated "subdocuments" corresponding to M different media types. $D = \left(d^1, d^2, \ldots, d^M\right)$, where d^1 is text, d^2 is image, d^M is the M-th media type.*

7.3 Methodology for Text-Image Topic Discovery

Existing multi-view works cannot do this task very well. E.g., [39] heavily relies on similarity graph for each view and completely ignores the detailed information, and even doesn't explicitly give topic representation; [40] assigns large weight to the domain with the largest approximation error and its performance will be dominated by the worst domain. We use matrix norm based numerical optimization instead of probabilistic graphical model (PGM) because it is difficult to design an accurate PGM to model the structure of images. Meanwhile, outliers and noisy terms usually degrade the performance, we thus use $l_{2,1}$-norm to learn robust topics. Also, by regularizing on the image graph, two text vectors with similar topic indicators should have higher similarity computed from other media types. In this way, multiple media type information can be effectively used and mutually enhanced to get the final consistent and representative text-image topics. We thus propose a novel regularized nonnegative constrained $l_{2,1}$ norm minimization (RNL21NM) framework to tackle our task:

$$\min_{\mathbf{F}\geq 0,\mathbf{G}\geq 0} \quad \left\|\mathbf{X}-\mathbf{G}\mathbf{F}^T\right\|_{2,1} + \frac{\lambda}{2}\sum_{m=2}^{M}\sum_{i,j}\alpha_m S_{ij}^m \left\|\mathbf{g}^i-\mathbf{g}^j\right\|_2^2 + \nu\|\mathbf{F}\|_1. \tag{7.1}$$

• $\mathbf{X} = [\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_N}] \in \mathcal{R}_+^{\mathbf{N}\times\mathbf{d}}$ is a nonnegative text data matrix, each row of $\mathbf{X}$ corresponds to a text subdocument;

• $\mathbf{F} \in \mathcal{R}_+^{\mathbf{d}\times\mathbf{K}}$ is a nonnegative topic basis matrix (one topic per column), K is the number of text-image topics;

• $\mathbf{G} \in \mathcal{R}_+^{\mathbf{N}\times\mathbf{K}}$ is a nonnegative topic indicator matrix (the i-th row $\mathbf{g}^i$ is the topic indicator vector for the i-th multimedia document), G_{ij} denotes the strength of the association between multimedia document D_i and text-image topic T_j;

• $\mathbf{S}^m \in \mathcal{R}_+^{N\times N}, m = 2, 3, \ldots, M$ is the similarity matrix for multimedia document dataset with respect to the m-th multimedia type (computation of $\mathbf{S}^m$ will be further discussed in the experiment section), α_m $s.t.$ $\sum_{m=2}^{M}\alpha_m = 1$ is the weight for the multimedia regularization term w.r.t. the m-th media type;

• $\lambda > 0$ is the multimedia regularization term which controls the impact of multimedia regularization on the convergent multimedia topic indicator matrix, the larger λ is, the more impact from the multimedia regularization is imposed.

• $\nu > 0$ controls the sparse regularization on the topic matrix $\mathbf{F}$.

The second term is the multimedia regularization term, which ensures that the multimedia topic indicator vectors $\mathbf{g}^i$ and $\mathbf{g}^j$ should be close if multimedia document D_i and multimedia document D_j are close in terms of multimedia topic concepts ($\mathbf{g}^i$ and $\mathbf{g}^j$ may not be necessarily far away if their corresponding multimedia documents are different since sometimes images of similar topics may have dissimilar visual feature representations).

We use the sum-absolute-value norm [92] to constrain matrix $\mathbf{F}$ in the third term to avoid trivial solutions[1] and make sparse representation of topics.

Let $\mathbf{S} = \sum_{m=2}^{M}\alpha_m\mathbf{S}^m$ be the integrated multimedia similarity matrix, where S_{ij} indicates the similarity between multimedia document D_i and multimedia document D_j. Note that $\mathbf{S}$ should be

[1] A trivial solution means that if $\mathbf{F}$ and $\mathbf{G}$ is a solution, then $a\mathbf{F}$ and $\frac{1}{a}\mathbf{G}$ for $a > 1$ is a better solution if without the regularization term in Eq. (7.1)

symmetric. Similar to dimensionality reduction via Laplacian eigenmap [93], we can further rewrite the multimedia regularization term into a succinct form as is shown in Eq. (7.2):

$$\begin{aligned} & \sum_i \sum_j S_{ij} \left\| \mathbf{g}^i - \mathbf{g}^j \right\|_2^2 \\ = \; & 2 \sum_i \left(\sum_j S_{ij} \right) \mathbf{g}^i \mathbf{g}^{iT} - 2 \sum_i \sum_j S_{ij} \mathbf{g}^i \mathbf{g}^{jT} \\ = \; & 2\mathrm{Tr} \left[\mathbf{G}^{\mathrm{T}} \left(\mathbf{D} - \mathbf{S} \right) \mathbf{G} \right] \\ = \; & 2\mathrm{Tr} \left[\mathbf{G}^{\mathrm{T}} \mathbf{L} \mathbf{G} \right] \end{aligned} \tag{7.2}$$

where $\mathbf{L}$ is the Laplacian matrix induced by $\mathbf{S}$.

Substituting Eq. (7.2) for the multimedia regularization term in the original framework Eq. (7.1) gives the regularized nonnegative constrained $l_{2,1}$ norm minimization (RNL21NM):

$$\min_{\mathbf{F} \geq 0, \mathbf{G} \geq 0} \left\| \mathbf{X} - \mathbf{G}\mathbf{F}^T \right\|_{2,1} + \lambda \mathrm{Tr} \left[\mathbf{G}^{\mathbf{T}} \mathbf{L} \mathbf{G} \right] + \nu \left\| \mathbf{F} \right\|_1 , \tag{7.3}$$

where

$$\mathbf{L} = \sum_{m=2}^{M} \alpha_m \mathbf{L}^m \tag{7.4}$$

is the multimedia Laplacian matrix, which is a linear combination of $M-1$ components. Each component corresponds to Laplacian matrix for a media type (i.e., image or video), where α_m, $\sum_{m=2}^{M} \alpha_m = 1$, is the weight on the m-th media type, $\mathbf{L}^m$ is the Laplacian matrix for the mth media type given by $\mathbf{L}^m = \mathbf{D}^m - \mathbf{S}^m$ where $\mathbf{S}^m$ is a similarity matrix w.r.t. the m-th media type and $\mathbf{D}^m$ is the diagonal matrix given by $D^m_{ii} = \sum_{j=1}^{N} S^m_{ij}$. Sometimes we use the normalized Laplacian defined by $\tilde{\mathbf{L}}^m = (\mathbf{D}^m)^{-1/2} \mathbf{L}^m (\mathbf{D}^m)^{-1/2}$. λ is a regularization parameter, which can be set based on confidence on the similarity obtained by multimedia information.

Note that there are several similar but substantially different existing formulations in literature such as NMF [3] (including its extensions) and Robust NMF [62]. NMFs use Frobenius norm or Kullback-Leibler divergence to minimize the approximation error. Since $l_{2,1}$-norm is not smooth, its optimization problem is more difficult than traditional NMFs. Although Robust NMF uses $l_{2,1}$-norm, it doesn't have regularization terms. However, adding regularization terms makes the

optimization problem more difficult, and the algorithm derived in [62] cannot be directly applied to solve problem (7.1). As will be shown later, we propose a new algorithm that can solve both problem (7.1) and Robust NMF.

7.3.1 Optimization Algorithm

In this section, we present a simple and efficient iterative algorithm to solve problem (7.3). The $l_{2,1}$-norm term is non-smooth and the objective function is not convex w.r.t. $\mathbf{G}$ and $\mathbf{F}$ simultaneously. We alternatively update one while keeping the other one fixed. We thus have the following two subproblems:

- Fix $\mathbf{F}$, update $\mathbf{G}$:

$$\min_{\mathbf{G}\geq\mathbf{0}} \left\|\mathbf{X}-\mathbf{G}\mathbf{F}^{T}\right\|_{2,1}+\lambda \mathrm{Tr}\left[\mathbf{G}^{\mathrm{T}}\mathbf{L}\mathbf{G}\right]. \tag{7.5}$$

- Fix $\mathbf{G}$, update $\mathbf{F}$:

$$\min_{\mathbf{F}\geq\mathbf{0}} \left\|\mathbf{X}-\mathbf{G}\mathbf{F}^{T}\right\|_{2,1}+\nu\|\mathbf{F}\|_{1}. \tag{7.6}$$

Although there are already several papers [21][62] on optimizing $l_{2,1}$-norm, their updating rules cannot be adapted to solve our problem. For example, [21] doesn't impose nonnegativity constraint, whereas [62] doesn't consider $l_{2,1}$-norm with regularization. Additionally, the optimization technique leveraging a well designed auxiliary function in [62] is slow, because there is always a nonnegative gap between the auxiliary function and the objective function and the auxiliary function doesn't utilize second order information either. In this section, we propose a new efficient algorithm to solve the regularized $l_{2,1}$-norm minimization problem.

We first prove a proposition that is important for optimizing problem (7.5) and (7.6).

Proposition 7.3.1. *If* $\forall \mathbf{X}_t\ , \mathbf{X}_t \in \arg\max_{\mathbf{X}}\{f\left(\mathbf{X}\right)-g\left(\mathbf{X};\mathbf{X}_t\right)\}$, *then*

$$g\left(\mathbf{X}_{t+1};\mathbf{X}_t\right)\leq g\left(\mathbf{X}_t;\mathbf{X}_t\right)\Rightarrow f\left(\mathbf{X}_{t+1}\right)\leq f\left(\mathbf{X}_t\right),$$

which further implies

$$
\begin{aligned}
& g\left(\mathbf{X}_{t+1};\mathbf{X}_t\right)+h\left(\mathbf{X}_{t+1}\right)\leq g\left(\mathbf{X}_t;\mathbf{X}_t\right)+h\left(\mathbf{X}_t\right), \\
\Rightarrow \qquad & f\left(\mathbf{X}_{t+1}\right)+h\left(\mathbf{X}_{t+1}\right)\leq f\left(\mathbf{X}_t\right)+h\left(\mathbf{X}_t\right).
\end{aligned}
$$

Proof.

$$
\begin{aligned}
& \because \mathbf{X}_t\in\arg\max_{\mathbf{X}} f\left(\mathbf{X}\right)-g\left(\mathbf{X};\mathbf{X}_t\right),\ \forall\mathbf{X}_t, \\
& \therefore f\left(\mathbf{X}_{t+1}\right)-g\left(\mathbf{X}_{t+1};\mathbf{X}_t\right)\leq f\left(\mathbf{X}_t\right)-g\left(\mathbf{X}_t;\mathbf{X}_t\right),\ \forall\mathbf{X}_t.
\end{aligned}
$$

If $g\left(\mathbf{X}_{t+1};\mathbf{X}_t\right)\leq g\left(\mathbf{X}_t;\mathbf{X}_t\right)$, then $f\left(\mathbf{X}_{t+1}\right)\leq f\left(\mathbf{X}_t\right)$. This implies that if $g\left(\mathbf{X}_{t+1};\mathbf{X}_t\right)+h\left(\mathbf{X}_{t+1}\right)\leq g\left(\mathbf{X}_t;\mathbf{X}_t\right)+h\left(\mathbf{X}_t\right)$, then $f\left(\mathbf{X}_{t+1}\right)+h\left(\mathbf{X}_{t+1}\right)\leq f\left(\mathbf{X}_t\right)+h\left(\mathbf{X}_t\right)$. □

Let $\gamma\left(x\right)=x-\frac{x^2}{2a}, a>0$, from concavity of $\gamma\left(x\right)$ we have $\gamma\left(x\right)\leq\gamma\left(a\right)+\left\langle\partial\gamma\left(a\right),x-a\right\rangle,\ \forall x$. We thus have $b-\frac{b^2}{2a}\leq a-\frac{a^2}{2a},\quad \forall b\in\mathcal{R}, a>0$.

Let $f\left(\mathbf{G}\right)=\sum_{i=1}^{N}\left\|\mathbf{x}^i-\mathbf{g}^i\mathbf{F}^T\right\|_2$, $h\left(\mathbf{G}\right)=\lambda\mathrm{Tr}\left[\mathbf{G}^{\mathrm{T}}\mathbf{L}\mathbf{G}\right]$, and $g\left(\mathbf{G};\mathbf{G}_t\right)=\sum_{i=1}^{N}\frac{\left\|\mathbf{x}^i-\mathbf{g}^i\mathbf{F}^T\right\|_2^2}{2\left\|\mathbf{x}^i-\mathbf{g}_t^i\mathbf{F}^T\right\|_2}$, we have

$$
\mathbf{G}_t\in\arg\max_{\mathbf{G}}\left\{f\left(\mathbf{G}\right)-g\left(\mathbf{G};\mathbf{G}_t\right)\right\},\ \forall\mathbf{G}_t.
$$

By Proposition 7.3.1, if $\mathbf{G}_{t+1}=\arg\min\left\{g\left(\mathbf{G};\mathbf{G}_t\right)+h\left(\mathbf{G}\right)\right\}$, we must have $g\left(\mathbf{G}_{t+1};\mathbf{G}_t\right)+h\left(\mathbf{G}_{t+1}\right)\leq g\left(\mathbf{G}_t;\mathbf{G}_t\right)+h\left(\mathbf{G}_t\right)$, i.e.,

$$
\left\|\mathbf{X}-\mathbf{G}_{t+1}\mathbf{F}^T\right\|_{2,1}+\lambda\mathrm{Tr}\left[\mathbf{G}_{\mathrm{t+1}}^{\mathrm{T}}\mathbf{L}\mathbf{G}_{\mathrm{t+1}}\right]\leq\left\|\mathbf{X}-\mathbf{G}_{\mathrm{t}}\mathbf{F}^{\mathrm{T}}\right\|_{2,1}+\lambda\mathrm{Tr}\left[\mathbf{G}_{\mathrm{t}}^{\mathrm{T}}\mathbf{L}\mathbf{G}_{\mathrm{t}}\right]. \tag{7.7}
$$

From the previous reasoning, we only need to solve the following optimization problem to update $\mathbf{G}$:

$$
\min_{\mathbf{G}\geq\mathbf{0}}\sum_{i=1}^{N}\frac{\left\|\mathbf{x}^i-\mathbf{g}^i\mathbf{F}^T\right\|_2^2}{\left\|\mathbf{x}^i-\mathbf{g}_l^i\mathbf{F}^T\right\|_2}+2\lambda\mathrm{Tr}\left[\mathbf{G}^{\mathrm{T}}\mathbf{L}\mathbf{G}\right]. \tag{7.8}
$$

which is equivalent to

$$\min_{\mathbf{G}\geq\mathbf{0}} \left\|\mathbf{DX}-\mathbf{DGF}^{T}\right\|_{F}^{2}+2\lambda\mathrm{Tr}\left[\mathbf{G}^{\mathrm{T}}\mathbf{LG}\right], \tag{7.9}$$

where $\mathbf{D}$ is a diagonal matrix with $D_{ii}=\left\|\mathbf{x}^{i}-\mathbf{g}_{t}^{i}\mathbf{F}^{T}\right\|_{2}^{-\frac{1}{2}}$.

Let $\mathcal{O}(\mathbf{G})$ be the objective function of $\mathbf{G}$ given fixed $\mathbf{F}$ in problem (7.9)

$$\mathcal{O}\left(\mathbf{G}\right)=\mathrm{Tr}[\mathbf{D}\left(\mathbf{X}-\mathbf{GF}^{\mathrm{T}}\right)\left(\mathbf{X}-\mathbf{GF}^{\mathrm{T}}\right)^{\mathrm{T}}\mathbf{D}]+2\lambda\mathrm{Tr}[\mathbf{G}^{\mathrm{T}}\mathbf{LG}],$$

the first derivative of $\mathcal{O}(\mathbf{G})$ with respect to G_{ij} is

$$\frac{\partial\mathcal{O}\left(\mathbf{G}\right)}{\partial G_{ij}}=2\left[\mathbf{D}^{2}\mathbf{GF}^{T}\mathbf{F}-\mathbf{D}^{2}\mathbf{XF}\right]_{ij}+4\lambda[\mathbf{LG}]_{ij},$$

and the second derivative of $\mathcal{O}(\mathbf{G})$ with respect to G_{ij} is

$$\mathcal{O}''\left(G_{ij}^{t}\right)=2D_{ii}^{2}\left[\mathbf{F}^{T}\mathbf{F}\right]_{jj}+4\lambda L_{ii}.$$

In each step, we intend to find the optimal G_{ij} in the sub-problem of Eq. (7.5) instead of find a better point leveraging an auxiliary function, as is shown below.

$$G_{ij}^{t+1}=\arg\min_{G_{ij}\geq0}\mathcal{O}\left(G_{ij}\right). \tag{7.10}$$

Since $\mathcal{O}(G_{ij})$ is a convex quadratic function given fixed $\mathbf{F}$, we could accurately represent $\mathcal{O}(G_{ij})$ as its second order Tyler expansion at last solution point $\mathbf{G}_t$ by

$$\begin{aligned}
& \mathcal{O}\left(G_{ij}\right)\\
= \;& \mathcal{O}\left(G_{ij}^{t}\right)+\mathcal{O}'\left(G_{ij}^{t}\right)\left(G_{ij}-G_{ij}^{t}\right)+\frac{1}{2}\mathcal{O}''\left(G_{ij}^{t}\right)\left(G_{ij}-G_{ij}^{t}\right)^{2}\\
= \;& \mathcal{O}\left(G_{ij}^{t}\right)+\frac{1}{2}\mathcal{O}''\left(G_{ij}^{t}\right)\left[G_{ij}-\left(G_{ij}^{t}-\frac{\mathcal{O}'\left(G_{ij}^{t}\right)}{\mathcal{O}''\left(G_{ij}^{t}\right)}\right)\right]^{2}-\frac{1}{2}\left(\frac{\mathcal{O}'\left(G_{ij}^{t}\right)}{\mathcal{O}''\left(G_{ij}^{t}\right)}\right)^{2}\mathcal{O}''\left(G_{ij}^{t}\right).
\end{aligned} \tag{7.11}$$

It can be shown that the optimizer of G_{ij} is

$$G_{ij}^{t+1} = \arg\min_{G_{ij}\geq 0} \mathcal{O}\left(G_{ij}\right) = \max\left(G_{ij}^{t} - \frac{\mathcal{O}'\left(G_{ij}^{t}\right)}{\mathcal{O}''\left(G_{ij}^{t}\right)}, 0\right),$$

giving the updating rules:

$$G_{ij} \leftarrow \max\left(0, G_{ij} - \frac{\left[\mathbf{D}^2\mathbf{G}\mathbf{F}^T\mathbf{F} - \mathbf{D}^2\mathbf{X}\mathbf{F} + 2\lambda\mathbf{L}\mathbf{G}\right]_{ij}}{D_{ii}^2[\mathbf{F}^T\mathbf{F}]_{jj} + 2\lambda L_{ii}}\right).$$

The procedure to update $\mathbf{G}$ is listed in Algorithm 8.

Algorithm 8 Update-G

Input: $\mathbf{X}, \mathbf{F}, \lambda, \mathbf{L}, \mathbf{D}, \mathbf{G}_t \in \mathcal{R}^{N\times K}$,
Output: $\mathbf{G}_{t+1}$.
$\mathbf{G} \leftarrow \mathbf{G}_t$
$\mathbf{A} \leftarrow \mathbf{F}^T\mathbf{F}$
$\mathbf{B} \leftarrow -\mathbf{D}^2\mathbf{X}\mathbf{F}$
$\mathbf{Q} \leftarrow diag\left(\mathbf{D}^2\right) diag\left(\mathbf{F}^T\mathbf{F}\right)^T + 2\lambda diag\left(\mathbf{L}\right)\mathbf{1}_{1\times K}$
repeat
  for $i = 1$ to N **do**
    for $j = 1$ to K **do**
      $G_{ij} \leftarrow \max\left(0, G_{ij} - \frac{D_{ii}^2\mathbf{G}_{i:}\mathbf{A}_{:j} + B_{ij} + 2\lambda\mathbf{L}_{i:}\mathbf{G}_{:j}}{D_{ii}^2 A_{jj} + 2\lambda L_{ii}}\right)$
    end for
  end for
until Convergence criterion satisfied
$\mathbf{G}_{t+1} \leftarrow \mathbf{G}$

Similarly, let $f\left(\mathbf{F}\right) = \sum_{i=1}^{N}\left\|\mathbf{x}^i - \mathbf{g}^i\mathbf{F}^T\right\|_2$, $h\left(\mathbf{F}\right) = \nu\|\mathbf{F}\|_1$, and $g\left(\mathbf{F};\mathbf{F}_t\right) = \sum_{i=1}^{N}\frac{\left\|\mathbf{x}^i - \mathbf{g}^i\mathbf{F}^T\right\|_2^2}{2\left\|\mathbf{x}^i - \mathbf{g}^i\mathbf{F}_t^T\right\|_2}$, we have $\mathbf{F}_t \in \arg\max_{\mathbf{F}}\left\{f\left(\mathbf{F}\right) - g\left(\mathbf{F};\mathbf{F}_t\right)\right\}$, $\forall\mathbf{F}_t$. By Proposition 7.3.1, if $\mathbf{F}_{t+1} = \arg\min\left\{g\left(\mathbf{F};\mathbf{F}_t\right) + h\left(\mathbf{F}\right)\right\}$, we must have $g\left(\mathbf{F}_{t+1};\mathbf{F}_t\right) + h\left(\mathbf{F}_{t+1}\right) \leq g\left(\mathbf{F}_t;\mathbf{F}_t\right) + h\left(\mathbf{F}_t\right)$, i.e.,

$$\left\|\mathbf{X} - \mathbf{G}\mathbf{F}_{t+1}^T\right\|_{2,1} + \nu\|\mathbf{F}_{t+1}\|_1 \leq \left\|\mathbf{X} - \mathbf{G}\mathbf{F}_t^T\right\|_{2,1} + \nu\|\mathbf{F}_t\|_1. \tag{7.12}$$

The procedure to compute $\mathbf{F}$ can thus be obtained by optimizing the following problem:

$$\min_{\mathbf{F}\geq\mathbf{0}}\left\|\mathbf{D}\mathbf{X} - \mathbf{D}\mathbf{G}\mathbf{F}^T\right\|_F^2 + 2\nu\|\mathbf{F}\|_{sav}, \tag{7.13}$$

where $\mathbf{D}$ is a diagonal matrix with $D_{ii} = \left\|\mathbf{x}^i - \mathbf{g}^i\mathbf{F}_t^T\right\|_2^{-\frac{1}{2}}$, and the updating rule for $\mathbf{F}$ is

$$F_{ij} \leftarrow \max\left(0, F_{ij} - \frac{\left[\mathbf{F}\mathbf{G}^T\mathbf{D}^2\mathbf{G} - \mathbf{X}^T\mathbf{D}^2\mathbf{G}\right]_{ij} + \nu}{\left[\mathbf{G}^T\mathbf{D}^2\mathbf{G}\right]_{jj}}\right).$$

Since the rows of $\mathbf{F}$ are mutually independent in Problem (7.13), we can update one column simultaneously at one time, which can greatly speed up the computation for RNL21NM, as is shown below:

$$\mathbf{F}_{:j} \leftarrow \max\left(\mathbf{0}, \mathbf{F}_{:j} - \frac{\mathbf{F}\left[\mathbf{G}^T\mathbf{D}^2\mathbf{G}\right]_{:j} - \left[\mathbf{X}^T\mathbf{D}^2\mathbf{G}\right]_{:j} + \nu}{\left[\mathbf{G}^T\mathbf{D}^2\mathbf{G}\right]_{jj}}\right).$$

The procedure to update $\mathbf{F}$ is listed in Algorithm 9. Ultimately, the algorithm to solve RNL21NM

Algorithm 9 Update-F

Input: $\mathbf{X}, \mathbf{G}, \nu, \mathbf{D}, \mathbf{F}_t \in \mathcal{R}^{d\times K}$,
Output: $\mathbf{F}_{t+1}$.
$\mathbf{F} \leftarrow \mathbf{F}_t$
$\mathbf{A} \leftarrow \mathbf{G}^T\mathbf{D}^2\mathbf{G}$
$\mathbf{B} \leftarrow -\mathbf{X}^T\mathbf{D}^2\mathbf{G}$
repeat
  for $j = 1$ to K **do**
    $\mathbf{F}_{:j} \leftarrow \max\left(\mathbf{0}, \mathbf{F}_{:j} - \frac{\mathbf{F}\mathbf{A}_{:j}+\mathbf{B}_{:j}+\nu}{A_{jj}}\right)$
  end for
until Convergence criterion satisfied
$\mathbf{F}_{t+1} \leftarrow \mathbf{F}$

is listed in Algorithm 10. We now prove the convergence of the proposed iterative procedure in Algorithm 10.

Theorem 7.3.2. *The alternative procedure in Algorithm 10 monotonically decrease the objective function value of problem (7.3).*

Proof. Denote

$$\mathscr{L}(\mathbf{G}, \mathbf{F}) \triangleq \left\|\mathbf{X} - \mathbf{G}\mathbf{F}^T\right\|_{2,1} + \lambda \mathrm{Tr}\left[\mathbf{G}^{\mathrm{T}}\mathbf{L}\mathbf{G}\right] + \nu\|\mathbf{F}\|_1, \tag{7.14}$$

according to Inequality (7.7) and (7.12), we have

$$\mathscr{L}(\mathbf{G}_{t+1}, \mathbf{F}_{t+1}) \leq \mathscr{L}(\mathbf{G}_{t+1}, \mathbf{F}_t) \leq \mathscr{L}(\mathbf{G}_t, \mathbf{F}_t).$$

Algorithm 10 RNL21NM

Input: $\mathbf{X}, \mathbf{L}, \lambda, \nu, \mathbf{G}_0, \mathbf{F}_0$,
Output: $\mathbf{G}, \mathbf{F}$.
$t \leftarrow 0$
repeat

$$\mathbf{D} \leftarrow \begin{bmatrix} \left\|\mathbf{x}^1 - \mathbf{g}_t^1 \mathbf{F}_t^T\right\|_2^{-\frac{1}{2}} & & \\ & \ddots & \\ & & \left\|\mathbf{x}^N - \mathbf{g}_t^N \mathbf{F}_t^T\right\|_2^{-\frac{1}{2}} \end{bmatrix}$$

$\mathbf{G}_{t+1} \leftarrow$ Update-G$(\mathbf{X}, \mathbf{F}_t, \lambda, \mathbf{L}, \mathbf{D}, \mathbf{G}_t)$

$$\mathbf{D} \leftarrow \begin{bmatrix} \left\|\mathbf{x}^1 - \mathbf{g}_{t+1}^1 \mathbf{F}_t^T\right\|_2^{-\frac{1}{2}} & & \\ & \ddots & \\ & & \left\|\mathbf{x}^N - \mathbf{g}_{t+1}^N \mathbf{F}_t^T\right\|_2^{-\frac{1}{2}} \end{bmatrix}$$

$\mathbf{F}_{t+1} \leftarrow$ Update-F$(\mathbf{X}, \mathbf{G}_{t+1}, \nu, \mathbf{D}, \mathbf{F}_t)$
$t \leftarrow t + 1$
until Convergence criterion satisfied
$\mathbf{G} \leftarrow \mathbf{G}_t$
$\mathbf{F} \leftarrow \mathbf{F}_t$

Since $\mathcal{R}$ is complete and $\{\mathscr{L}(\mathbf{G}_t, \mathbf{F}_t)\}$ is bounded from below by 0, the sequence converges to its infimum. □

7.3.2 Stopping Condition

One common choice of stopping condition for bound constrained optimization is to test whether the norm of the projected gradient is less than a fixed tolerance as in [56][61]. Let T be a projection operator on the nonnegative orthant as is defined by

$$[T_{\mathbf{X}}\mathbf{M}]_{ij} = \begin{cases} M_{ij} & \text{if} \quad X_{ij} > 0 \\ \min\{M_{ij}, 0\} & \text{if} \quad X_{ij} = 0 \end{cases}. \tag{7.15}$$

The projected gradient for $\mathscr{L}(\mathbf{G}, \mathbf{F})$ in the objective function in Problem (7.3) w.r.t. $\mathbf{G}$ and $\mathbf{F}$ are defined by

$$P\nabla\mathbf{G} \triangleq T_{\mathbf{G}}\nabla_{\mathbf{G}}\mathcal{L}(\mathbf{G}, \mathbf{F}), \; P\nabla\mathbf{F} \triangleq T_{\mathbf{F}}\nabla_{\mathbf{F}}\mathcal{L}(\mathbf{G}, \mathbf{F}).$$

According to KKT conditions, $(\mathbf{G}^*, \mathbf{F}^*)$ is an optimal solution if and only if $(P\nabla\mathbf{G}, P\nabla\mathbf{F}) = \mathbf{0}$, thus we can use the norm of project gradient to measure how close the current point is to the

optimizer. One commonly used convergence criterion is

$$\|[P\nabla \mathbf{G}; P\nabla \mathbf{F}]\|_F^2 \leq \varepsilon \|[\nabla_{\mathbf{G}} \mathcal{L}(\mathbf{G}_0, \mathbf{F}_0); \nabla_{\mathbf{F}} \mathcal{L}(\mathbf{G}_0, \mathbf{F}_0)]\|_F^2,$$

where $\mathbf{G}^0$ and $\mathbf{F}^0$ are starting points.

7.3.3 Computation Complexity

In each outer iteration of calculating $\mathbf{G}$, we need to calculate $\mathbf{XF}$ and $\mathbf{F}^T\mathbf{F}$ in $O(dNK)$ and $O(dK^2)$ respectively in advance. During each inner iteration, updating $\mathbf{G}$ costs $O(k_{nn}NK)$ where k_{nn} is the neighborhood size on the sparse data graph (Since each row of the Laplacian Matrix $\mathbf{L}$ has only $O(k_{nn})$ non-zero elements, the term $\mathbf{LG}$ can be computed efficiently), therefore, the complexity of updating $\mathbf{G}$ in Algorithm 8 is $O(dNK) + \#\text{sub-iterations} \times O(k_{nn}NK)$. The computation of $\mathbf{F}$ can be much faster since we update a column at one time. Its computational complexity is $O(dNK) + \#\text{sub-iterations} \times O(dK^2)$. In summary, the total computational complexity for Algorithm 10 is $\#\text{iters} \times \left(O(dNK) + \#\text{sub-iters} \times O(dK^2 + k_{nn}NK)\right)$, where N is the number of documents, and d is the vocabulary size. The memory cost is $O(Nk_{nn}) + O(NK) + O(dK)$.

Since the computational complexity and memory cost of RNL21NM is linear to the feature size d and the data size N, the proposed method can be run on big data. The only restriction is the requirement that data and intermediate matrices should be stored in memory since it is a sequential and iterative algorithm. In this case, one can use Apache Spark to process big data as it supports cyclic data flow and in-memory computing.

7.4 Experiments

In this section, we conduct extensive experiments to evaluate RNL21NM for joint text-image topic discovery task on two crawled text-image news datasets. We use clustering metrics to evaluate the topic discovery performance, since the clustering metric can objectively and quantitatively measure how coherent documents within a topic cluster are and how close the predicted topic assignment is to the ground truth.

7.4.1 Datasets

Our formulation of the joint text-image topic mining problem assumes that each text-image document has rich text contents and a good image. We still rely on rich text parts because mining topics on images alone is difficult. In multimedia domain, there is a Flickr dataset which includes tagged images. However, the tags are so few that they are not suitable for mining topics. Also, the Flickr dataset aims at image retrieval evaluation, not built for joint text-image topic mining. There is also a Corel dataset, but it contains only images, no associated text with it. To the best of our knowledge, there is no existing dataset we can use, thus we have to create our own data sets.

We collected two text-image news datasets by crawling CNN top stories and National Public Radio (NPR) news from RSS feeds. We implement a RSS feeds crawler in Java. Titles, abstracts and text body contents are extracted as the text part, meanwhile, the image associated with the news text is stored as the image part for text-image document (Ads icons are filtered out). Text contents are stemmed using Java portStemmer [86], and we use normalized TFIDF to represent a text subdocument. For image features, we use 7 color features, and 5 textual features. Specifically, color features include RGB dominant color, HSV dominant color, RGB color moment, HSV color moment, RGB color histogram, HSV color histogram, color coherence vector [87]. Texture features comprise four Tamura textural features [88] (coarseness, contrast, directionality, line-likeness) and Gabor transform [89, 90]. The first dataset we collected is CNN top stories from Feb. 21st, 2011 to April 17th, 2011. There are 10 top news articles in average each day, and some web pages contains only videos, and some other pages provide no videos or images, finally 142 text-image pairs are collected. The crawled data are manually labeled into 10 categories, the topic distribution is listed in Table 7.1. The second dataset we built is NPR news articles from Apr. 7th, 2013 to May 7th, 2013. Unlike CNN top stories, we crawl multiple RSS feeds of NPR news, and each RSS feed has been organized to focus on one major theme or topic by NPR staff. For example, all articles listed in the "education" RSS feed is about education topic. This strategy of collecting news data provides us reliable ground truth. The topic distribution on NPR dataset is shown in Table 7.2. Table 7.3 shows more description information for CNN and NPR datasets.

Please be noted that there are only few news articles updated in the RSS feeds, and some web pages contains only videos, while some other pages provide no videos or images at all, which limits

Table 7.1: Topic distribution for CNN top story data

Topic	#	Topic	#
Egyptian protest	5	Gbagbo peacekeep	8
personal news, life	26	traveling	7
Syrian demonstration	12	nuclear leak in Japan	27
Israel, Iraq attack	8	Libyan rebel	14
Politics and economy	26	crimes, victims	9

Table 7.2: Topic distribution for NPR news dataset

Topic	#	Topic	#
arts_culture	218	economy	57
education	27	environment	43
politics	121	religion	21
sports	49	technology	67

the scale of the datasets used in the experiments. Nevertheless, the conclusions would still be expected to hold for large scale data for which we will give qualitative analysis in the end of the section.

7.4.2 Compared Methods

The main hypothesis we would like to test is that leveraging correlations in different types of data to "jointly" mine topics is more effective than using a two-stage approach to mine each type of data separately. To test this hypothesis, we compare our method with several baseline approaches representing the two-stage strategy.

• [Text first, then images]: We first do traditional topic mining on text data where topic-word matrix could be obtained. After the first step, we have the textual document clusters; each cluster corresponds to a topic. We then select the associated images with the textual documents for each topic as the image representatives for that topic.

• [Images first, then text]: We can also do image clustering first, and assume that each cluster corresponds to a specific topic. After the first step, we have a set of textual documents within each

Table 7.3: Datasets description

Datasets	#Term	#Sample	#Class	Sparsity
CNN	8682	142	10	0.03594
NPR	17692	603	8	0.01168

cluster. We then calculate the weighted term vector using TF-IDF weighting on the document set in each cluster as the weighted term representor for that latent topic.

We thus have three baseline methods for two-stage strategies, two of them are text-first approaches using K-means and LDA respectively followed by selecting associated images according to the first stage result. The third one is an image-first approach where we first do K-means on images and then get the weighted term from the associated documents for each topic cluster.

For multi-view methods, we compare the state-of-the-art co-trained multi-view spectral clustering
(CoTrainedMVSC)[39] and robust multi-view K-means clustering (RMKMC)[40].

We compare the best average NMI and ACC for 10 runs for all methods. For LDA[2], the best average performance is achieved by grid search with $\alpha = 50/T, \beta = 200/W$ as is suggested by the implementor, where T is the number of topics, and W is the vocabulary size. For CoTrainedMV, we use cosine similarity for text view and Gaussian normalized negative Euclidean distance for image view. For RMKMC, following its author's suggestion, we choose the best parameter γ^* by searching $log_{10}(\gamma)$ in the range from 0.1 to 2 with incremental step 0.2. For RNL21NM, we set $\alpha_2 = 1$ since only image is included. We fix ν to be 0.0001. To compute the image similarity matrix, we first calculate the negative Euclidean distance matrix and then normalize all the matrix entries into zero-one interval by Gaussian normalization. We tune λ by grid-search from $\{0, 0.2, 0.4, 0.8, 1.6, \cdots, 51.2\}$.

7.4.3 Results and Discussion

The first question we want to answer is whether the discovered image-text topics are meaningful. We demonstrate the 8 text-image topics discovered by RNL21NM on the NPR dataset in Figure 7.1 (for lack of space text-image topics on CNN top stories are not shown but the characteristic of the CNN results is similar). It can be clearly seen that most topics are very semantically coherent. Human can easily differentiate topics by the visual picture alone. For example, Topic 3 contains pictures of mountains and grass field, farm, vegetable, and trees, indicating this topic is related to environment. Topic 4 shows many pictures on basketball, so it's probably on sport. This is very

[2] http://psiexp.ss.uci.edu/research/programs_data/toolbox.htm

Table 7.4: Clustering Results. $*$ means statistically significance at the 0.05 level.

NMI% $\pm$ std

Dataset	Kmeans_Text	Kmeans_Image	LDA	RNMF	CoTrainedMV	RMKMC	RNL21NM
CNN	56.4 ± 7.0	22.3 ± 1.6	59.4 ± 2.5	57.7 ± 4.8	41.6 ± 1.2	32.7 ± 4.7	$\mathbf{68.8}^* \pm \mathbf{2.9}$
NPR	26.6 ± 3.2	3.8 ± 0.4	37.7 ± 3.2	38.8 ± 4.0	17.8 ± 2.3	4.2 ± 0.4	$\mathbf{39.6}^* \pm \mathbf{2.5}$

ACC% $\pm$ std

Dataset	Kmeans_Text	Kmeans_Image	LDA	RNMF	CoTrainedMV	RMKMC	RNL21NM
CNN	54.9 ± 6.5	25.1 ± 1.1	53.6 ± 2.7	58.0 ± 5.7	42.3 ± 5.0	37.3 ± 4.6	$\mathbf{68.7}^* \pm \mathbf{5.3}$
NPR	39.2 ± 5.6	19.8 ± 1.8	48.4 ± 4.5	56.4 ± 7.9	37.1 ± 3.4	18.6 ± 2.3	$\mathbf{58.7}^* \pm \mathbf{3.3}$

convenient for people who don't read text terms, which is one of the advantages of joint text-image topics over traditional text term based representation. We also see that RNL21NM can discovery almost all classes in ground truth without any supervision. From the text-image topics shown in Figure 7.1, one can easily say that topic 1 is on politics, topic 2 is on economy, topic 3 environment, topic 4 sport, topic 5 education, topic 6 legislation, topic 7 movie and art, topic 8 technology, only religion is missed, replaced by legislation. However, the ground truth shows the religion class contains only 21 samples, and RNL21NM does discover major topics.

The second question we need to answer is how well RNL21NM works for the joint text-image topic discovery problem in terms of quantitative and objective metrics. We compare the average NMI and ACC in 10 runs for all methods in Table 7.4. The clustering results show that the proposed RNL21NM outperforms all the other methods. The results also show that image-first approach performs badly. This is not beyond of our expectation. There is a larger gap between image features and semantic concepts compared to that of text. The reason maybe that the features we used in this work are not good for capturing the semantics. However, image clustering performance is dramatically improved by using two-stage text-first mining methods and can be further improved by our unified approach, demonstrating that text information is quite helpful for image clustering especially when image features are not good. We also see that CoTrainedMVSC behaves fairly in the joint text-image topic mining task. CoTrainedMVSC's performance depends on high level similarity matrix from different views. However, how to define good similarity measure for different views is an important research problem itself. The difference between RNL21NM and CoTrainedMVSC is that RNL21NM utilizes the vector space model to analyze texts from the term-level whereas CoTrainedMVSC highly relies on the similarity matrix for the text part. Therefore RNL21NM uses detailed text information but CoTrainedMVSC does not. Although

CoTrainedMVSC behaves better than image-first K-means, it performs even worse than text-first K-means since CoTrainedMVSC tries to find consistent topic assignments across both text and image views, but the poor quality of image features severely degrades CoTrainedMVSC's performance. Also, RMKMC is vulnerable to the poor feature quality in the image domain.

The next question we examine is how the regularization parameter λ affects the performance of RNL21NM. We found that image quality plays an important role on RNL21NM's performance improvement over that by using text alone. From the performance of Kmeans-Image on Table 7.4, we already see that the image quality of CNN is better than NPR in terms of NMI and ACC. Figure 7.2 shows consistent results again. The best NMI and ACC of RNL21NM by using both text and images are higher than those using single text by over 10 percentage points on CNN dataset, whereas the NMI and ACC increase on NPR dataset is less than 4 percentage points. Besides, Figure 7.2 shows that the turning points of NMI and ACC curves occur at an appropriate λ and large λs hurt the performance. It also shows that the optimal λ depends on the image quality in terms of NMI and ACC. The better the image quality is, the larger is the optimal λ. For practitioners, we suggest using a validation set with ground truth under an affordable cost to tune the parameters by e.g. grid search. Also as we have discussed, different users may label the data points differently, we may want to use some group-wise parameter tuning where we ask similar users to build the validation set to tune parameters that work best for this group of users.

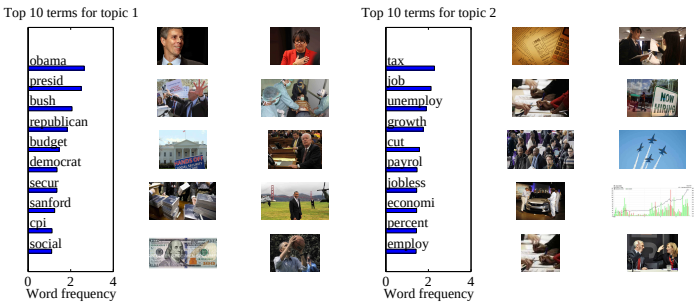
We finally study the convergence of RNL21NM. From Figure 7.3, we can see that the proposed optimization algorithm is effective and converges fast.

7.4.4 Limitation

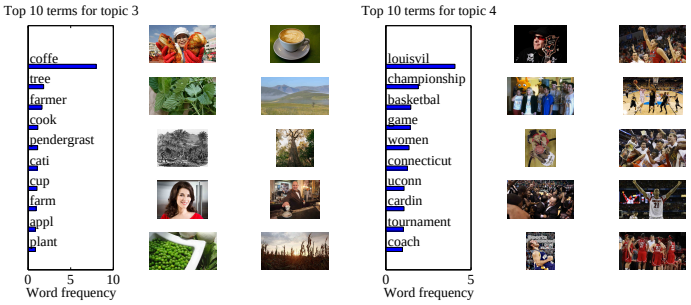
Please be noted that one limitation of the experiment design is that the scale of the crawled dataset is small thus the algorithms' performance rank might not be exactly the same on large scale datasets. But since the proposed method utilizes the detailed text information and we can tune the regularization parameter depending on the discriminative quality of the images, the advantage would still take effect for large scale web new datasets, therefore the proposed method would still be superior to the baseline methods for large scale web new data. Of course for quantitative validation, experiments on large-scale datasets need to be done before conclusions could be made.

7.5 Summary

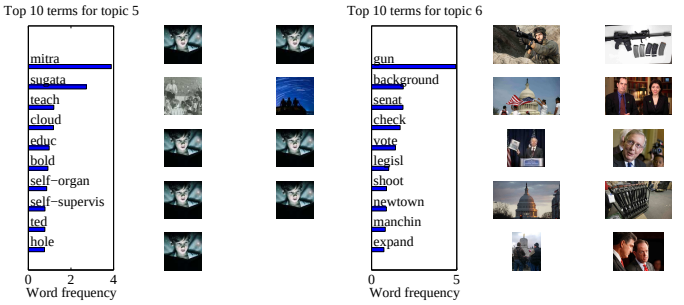
In this chapter, we define a new concept "text-image topics" and propose a general regularized nonnegative constrained $l_{2,1}$-norm minimization framework to discover text-image topics from web news collections by using not only text data but also data of other types such as images. We propose a novel iterative algorithm to solve the optimization problem. Experimental results on the crawled CNN and NPR web news datasets validate the efficacy of the proposed approach.



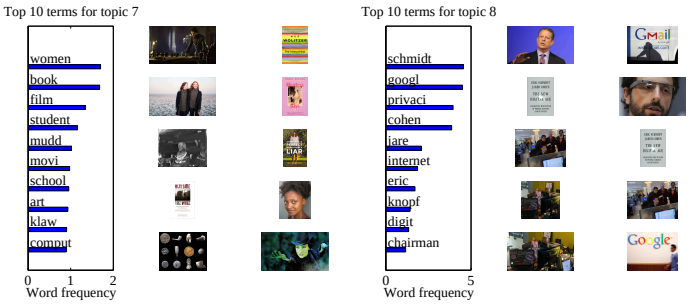
(a) NPR Topic 1 (b) NPR Topic 2



(c) NPR Topic 3 (d) NPR Topic 4

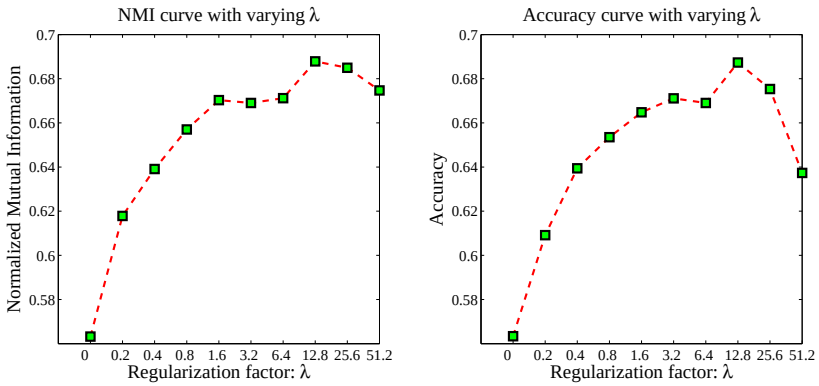


(e) NPR Topic 5 (f) NPR Topic 6

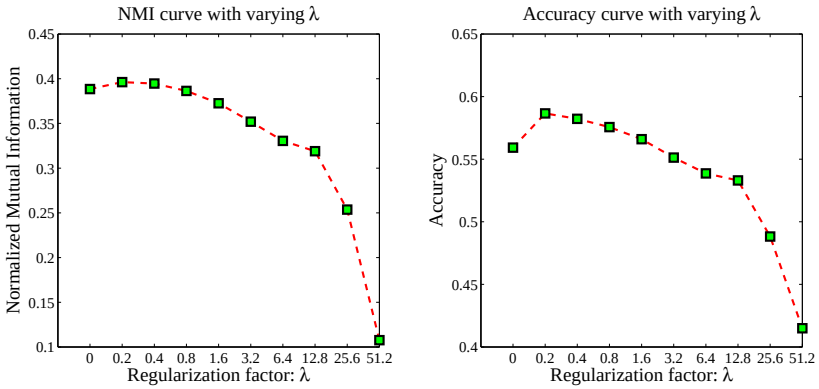


(g) NPR Topic 7 (h) NPR Topic 8

Figure 7.1: Text-image topics discovered by RNL21NM on the NPR web news dataset. For each text-image topic, we show the top 10 words in terms of weights, and the images associated with the topic. The text-image topics has better understandability since images are more understandable for people, and provides more vivid representation for the concept contained in the topic.



(a) CNN



(b) NPR

Figure 7.2: NMI and ACC curves for RNL21NM with varying λ.

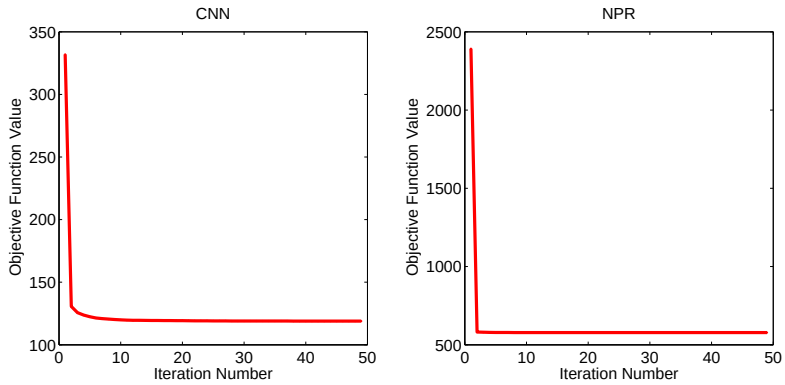


Figure 7.3: Convergence curve of RNL21NM on CNN and NPR datasets.

Chapter 8

Conclusions and Future Work

In this thesis four published works during the Ph.D. study of the author are discussed on single/multi-view unsupervised feature selection and multi-view topic discovery for text-image web news data.

For single-view unsupervised feature selection, we propose two novel methods RUFS and AUFS. RUFS considers outliers in both labeling learning and feature selection thus is more robust than state-of-the-arts. AUFS is proposed such that three desirable properties are satisfied: (1) The feature selection function should have the sparsity-inducing property; (2) It should equally penalize large weights and small weights, leading to a fair competition between different features; (3) The fitting term should achieve a good balance between small loss on normal data examples and large loss on outliers.

For multi-view unsupervised feature selection, we propose to directly utilize raw features in the main view (e.g., text for text-image web news data) to learn pseudo cluster labels which should also have the most consensus with other views (e.g., image), and meanwhile the discriminative features in the feature selection process will win out to contribute more on label learning process, and in return the improved cluster labels will help to select more discriminative features for each view.

For multi-view topic discovery, we propose a regularized nonnegative constrained $l_{2,1}$-norm minimization framework as a systematic solution that can integrate information propagation and mutual enhancement between data of different types without supervision in a principled way.

In a nutshell, the basic finding is that without label information it is still possible to learn the intrinsic patterns and by leveraging these patterns an effective and relevant feature subset can be automatically learned. What's more interesting, the selected features does not only improve clustering performance, more surprisingly, it indeed helps improve the classification accuracy for many public benchmark datasets as well, though the improvement on some datasets is only marginal.

Actually this phenomenon is often seen when a baby learns to recognize objects. Though without a teacher, the baby still manages to differentiate objects by the visual features. In this sense, classification can be viewed as labeled clustering or assigning label to a nearest cluster.

Of course there are limitations of our contributions. First, the proposed algorithms are all batch algorithms, which would encounter problems for data streams. In such case, online/incremental algorithms for unsupervised feature selection and topic discovery are more appropriate for scalability and practicability. Second, the evaluation of algorithms in all experiments in this thesis makes the assumption that the number of clusters for all methods should be set to the true number of clusters for posterior clustering algorithms. This makes sense because posterior clustering algorithms would after all determine the number of clusters, while for classification tasks the number of classes is a known constant. However, what if we treat the number of clusters as a variable? How will this variable affect the ultimate performance? Will the performance rank of different algorithms vary with respect to this variable? Unfortunately, to the best of our knowledge, few work has been done in this direction, which motivates us to study how to tune the number of clusters as a parameter in a principled way.

Inspired by that useful patterns can be learned in an unsupervised scenario, many applications can be potentially benefited by utilizing low-cost or even free large scale unlabeled data to select or construct features that are most relevant and effective to the learning tasks. For example, we can apply most/all of techniques presented in this thesis to new important application areas like electronic medical record (EMR) analysis. Specifically, we can study how to select a subset of features from large scale unlabeled data to improve heart failure survival score prediction. For recommender systems, a user may not click any item, but his or her browsing behavior may give some hint on his or her preferences if we have mined some typical patterns from a large number of unlabeled users and thereby select or construct effective features. For deep learning techniques, we usually feed the entire raw features into a learning system. But for some problems, we may add a filter to select most relevant subset of features in front of a deep network to expect significant speed-up or even improvement of accuracy for the learning system.

References

[1] T. Hofmann, "Probabilistic latent semantic indexing," in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 50–57, ACM, 1999.

[2] D. Blei, A. Ng, and M. Jordan, "Latent dirichlet allocation," *The Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[3] D. Lee and H. Seung, "Algorithms for non-negative matrix factorization," *Advances in neural information processing systems*, vol. 13, 2001.

[4] D. Cai, C. Zhang, and X. He, "Unsupervised feature selection for multi-cluster data," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 333–342, ACM, 2010.

[5] Y. Yang, H. Shen, Z. Ma, Z. Huang, and X. Zhou, "l 2, 1-norm regularized discriminative feature selection for unsupervised learning," in *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume Two*, pp. 1589–1594, AAAI Press, 2011.

[6] Z. Li, Y. Yang, J. Liu, X. Zhou, and H. Lu, "Unsupervised feature selection using nonnegative spectral analysis," in *Twenty-Sixth AAAI Conference on Artificial Intelligence (AAAI-12)*, pp. 1026–1032, AAAI Press, 2012.

[7] Y. Feng, J. Xiao, Y. Zhuang, and X. Liu, "Adaptive unsupervised multi-view feature selection for visual concept recognition," in *Proceedings of the 11th Asian conference on Computer Vision-Volume Part I*, pp. 343–357, Springer-Verlag, 2012.

[8] J. Tang, X. Hu, H. Gao, and H. Liu, "Unsupervised feature selection for multi-view data in social media," in *Proceedings of the 13th SIAM International Conference on Data Mining, 2013*, SIAM, 2013.

[9] M. A. Hall, *Correlation-based feature selection for machine learning*. PhD thesis, The University of Waikato, 1999.

[10] R. Duda, P. Hart, and D. Stork, "Pattern recognition. 2001,"

[11] X. He, D. Cai, and P. Niyogi, "Laplacian score for feature selection," *Advances in Neural Information Processing Systems*, vol. 18, pp. 507–514, 2006.

[12] Z. Zhao and H. Liu, "Spectral feature selection for supervised and unsupervised learning," in *Proceedings of the 24th international conference on Machine learning*, pp. 1151–1157, ACM, 2007.

[13] M. Masaeli, J. G. Dy, and G. M. Fung, "From transformation-based dimensionality reduction to feature selection," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pp. 751–758, 2010.

[14] H. Liu, X. Wu, and S. Zhang, "Feature selection using hierarchical feature clustering," in *Proceedings of the 20th ACM international conference on Information and knowledge management*, pp. 979–984, ACM, 2011.

[15] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artificial intelligence*, vol. 97, no. 1, pp. 273–324, 1997.

[16] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *The Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[17] A. Rakotomamonjy, "Variable selection using svm based criteria," *The Journal of Machine Learning Research*, vol. 3, pp. 1357–1370, 2003.

[18] V. Vapnik, *The nature of statistical learning theory*. springer, 1999.

[19] J. Zhu, S. Rosset, T. Hastie, and R. Tibshirani, "1-norm support vector machines," *Advances in neural information processing systems*, vol. 16, no. 1, pp. 49–56, 2004.

[20] C. Hou, F. Nie, D. Yi, and Y. Wu, "Feature selection via joint embedding learning and sparse regression," in *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume Two*, pp. 1324–1329, AAAI Press, 2011.

[21] F. Nie, H. Huang, X. Cai, and C. Ding, "Efficient and robust feature selection via joint l2, 1-norms minimization," *Advances in Neural Information Processing Systems*, vol. 23, pp. 1813–1821, 2010.

[22] Z. Zhao, L. Wang, and H. Liu, "Efficient spectral feature selection with minimum redundancy," in *Proceedings of the Twenty-4th AAAI Conference on Artificial Intelligence (AAAI)*, 2010.

[23] F. Nie, S. Xiang, Y. Jia, C. Zhang, and S. Yan, "Trace ratio criterion for feature selection," in *Proceedings of the 23rd national conference on Artificial intelligence*, vol. 2, pp. 671–676, 2008.

[24] K. Fukunaga, *Introduction to statistical pattern recognition*. Academic Pr, 1990.

[25] Q. Gu, Z. Li, and J. Han, "Joint feature selection and subspace learning," in *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume Two*, pp. 1294–1299, AAAI Press, 2011.

[26] T. Hofmann, "Unsupervised learning by probabilistic latent semantic analysis," *Machine Learning*, vol. 42, no. 1, pp. 177–196, 2001.

[27] C. Ding, T. Li, W. Peng, and H. Park, "Orthogonal nonnegative matrix t-factorizations for clustering," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 126–135, ACM, 2006.

[28] D. Blei and J. Lafferty, "Topic models," *Text Mining: Classification, Clustering, and Applications*, pp. 71–94.

[29] C. Zhai, "Statistical language models for information retrieval," *Synthesis Lectures on Human Language Technologies*, vol. 1, no. 1, pp. 1–141, 2008.

[30] R. Zhao and W. Grosky, "Narrowing the semantic gap-improved text-based web document retrieval using visual features," *Multimedia, IEEE Transactions on*, vol. 4, no. 2, pp. 189–200, 2002.

[31] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American society for information science*, vol. 41, no. 6, pp. 391–407, 1990.

[32] B. Gao, T. Liu, T. Qin, X. Zheng, Q. Cheng, and W. Ma, "Web image clustering by consistent utilization of visual features and surrounding texts," in *Proceedings of the 13th annual ACM international conference on Multimedia*, pp. 112–121, ACM, 2005.

[33] R. Bekkerman and J. Jeon, "Multi-modal clustering for multimedia collections," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pp. 1–8, IEEE, 2007.

[34] D. Mahajan and M. Slaney, "Image classification using the web graph," in *Proceedings of the international conference on Multimedia*, pp. 991–994, ACM, 2010.

[35] R. van Zwol, B. Sigurbjornsson, R. Adapala, L. Garcia Pueyo, A. Katiyar, K. Kurapati, M. Muralidharan, S. Muthu, V. Murdock, P. Ng, *et al.*, "Faceted exploration of image search results," in *Proceedings of the 19th international conference on World wide web*, pp. 961–970, ACM, 2010.

[36] X. Olivares, M. Ciaramita, and R. Van Zwol, "Boosting image retrieval through aggregating search results based on visual annotations," in *Proceeding of the 16th ACM international conference on Multimedia*, ACM, 2008.

[37] D. Blei and M. Jordan, "Modeling annotated data," in *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pp. 127–134, ACM, 2003.

[38] K. Barnard, P. Duygulu, D. Forsyth, N. De Freitas, D. Blei, and M. Jordan, "Matching words and pictures," *The Journal of Machine Learning Research*, vol. 3, pp. 1107–1135, 2003.

[39] A. Kumar and H. Daumé III, "A co-training approach for multi-view spectral clustering," in *Proceedings of the 28th annual international conference on machine learning*, 2011.

[40] X. Cai, F. Nie, and H. Huang, "Multi-view k-means clustering on big data," in *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, 2013.

[41] Y. Chen, L. Wang, and M. Dong, "Non-negative matrix factorization for semisupervised heterogeneous data coclustering," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 22, no. 10, pp. 1459–1474, 2010.

[42] L. Meng, A. Tan, and D. Xu, "Semi-supervised heterogeneous fusion for multimedia data co-clustering," *IEEE Transactions on Knowledge and Data Engineering*, 2013.

[43] C. Ding, T. Li, and M. Jordan, "Convex and semi-nonnegative matrix factorizations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 45–55, 2008.

[44] F. Wang, T. Li, and C. Zhang, "Semi-supervised clustering via matrix factorization," in *Proceedings of The 8th SIAM Conference on Data Mining*, Citeseer, 2008.

[45] D. Cai, X. He, X. Wu, and J. Han, "Non-negative matrix factorization on manifold," in *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pp. 63–72, IEEE, 2008.

[46] Q. Gu and J. Zhou, "Local learning regularized nonnegative matrix factorization," in *Proceedings of the 21st international jont conference on Artifical intelligence*, pp. 1046–1051, Morgan Kaufmann Publishers Inc., 2009.

[47] W. Xu, X. Liu, and Y. Gong, "Document clustering based on non-negative matrix factorization," in *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 267–273, ACM, 2003.

[48] F. Shahnaz, M. Berry, V. Pauca, and R. Plemmons, "Document clustering using nonnegative matrix factorization," *Information Processing & Management*, vol. 42, no. 2, pp. 373–386, 2006.

[49] C. Ding, X. He, and H. Simon, "On the equivalence of nonnegative matrix factorization and spectral clustering," in *Proc. SIAM Data Mining Conf*, no. 4, pp. 606–610, 2005.

[50] C. Ding, T. Li, and W. Peng, "Nonnegative matrix factorization and probabilistic latent semantic indexing: Equivalence chi-square statistic, and a hybrid method," in *Proceedings of the 21st National Conference on Artificial Intelligence*, vol. 21, p. 342, Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006.

[51] E. Gaussier and C. Goutte, "Relation between PLSA and NMF and implications," in *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 601–602, ACM, 2005.

[52] C. Ding, T. Li, and M. Jordan, "Convex and semi-nonnegative matrix factorizations," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 1, pp. 45–55, 2010.

[53] H. Ma, W. Zhao, Q. Tan, and Z. Shi, "Orthogonal nonnegative matrix tri-factorization for semi-supervised document co-clustering," *Advances in Knowledge Discovery and Data Mining*, pp. 189–200, 2010.

[54] P. Paatero and U. Tapper, "Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values," *Environmetrics*, vol. 5, no. 2, pp. 111–126, 1994.

[55] L. Grippo and M. Sciandrone, "On the convergence of the block nonlinear gauss-seidel method under convex constraints," *Operations Research Letters*, vol. 26, no. 3, pp. 127–136, 2000.

[56] C. Lin, "Projected gradient methods for nonnegative matrix factorization," *Neural computation*, vol. 19, no. 10, pp. 2756–2779, 2007.

[57] H. Kim and H. Park, "Non-negative matrix factorization based on alternating non-negativity constrained least squares and active set method," *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 2, pp. 713–730, 2008.

[58] D. Kim, S. Sra, and I. Dhillon, "Fast newton-type methods for the least squares nonnegative matrix approximation problem," in *Proceedings of the 2007 SIAM International Conference on Data Mining*, pp. 343–354, 2007.

[59] J. Kim and H. Park, "Toward faster nonnegative matrix factorization: A new algorithm and comparisons," in *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pp. 353–362, Ieee, 2008.

[60] A. Cichocki and A. Phan, "Fast local algorithms for large scale nonnegative matrix and tensor factorizations," *IEICE Transactions on Fundamentals of Electronics*, vol. 92, pp. 708–721, 2009.

[61] C. Hsieh and I. Dhillon, "Fast coordinate descent methods with variable selection for non-negative matrix factorization," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1064–1072, ACM, 2011.

[62] D. Kong, C. Ding, and H. Huang, "Robust nonnegative matrix factorization using l21-norm," in *Proceedings of the 20th ACM international conference on Information and knowledge management*, pp. 673–682, ACM, 2011.

[63] M. Wang, X. Hua, R. Hong, J. Tang, G. Qi, and Y. Song, "Unified video annotation via multigraph learning," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 19, no. 5, pp. 733–746, 2009.

[64] M. Wang, X. Hua, J. Tang, and R. Hong, "Beyond distance measurement: constructing neighborhood similarity for video annotation," *Multimedia, IEEE Transactions on*, vol. 11, no. 3, pp. 465–476, 2009.

[65] F. Nie, D. Xu, I.-H. Tsang, and C. Zhang, "Flexible manifold embedding: a framework for semi-supervised and unsupervised dimension reduction," *Image Processing, IEEE Transactions on*, vol. 19, no. 7, pp. 1921–1932, 2010.

[66] L. Bottou and V. Vapnik, "Local learning algorithms," *Neural computation*, vol. 4, no. 6, pp. 888–900, 1992.

[67] D. Lee, H. Seung, *et al.*, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[68] M. Wu and B. Scholkopf, "A local learning approach for clustering," *Advances in neural information processing systems*, vol. 19, p. 1529, 2007.

[69] J. Shi and J. Malik, "Normalized cuts and image segmentation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 8, pp. 888–905, 2000.

[70] J. Nocedal and S. Wright, *Numerical optimization*. Springer verlag, 1999.

[71] D. Liu and J. Nocedal, "On the limited memory bfgs method for large scale optimization," *Mathematical programming*, vol. 45, no. 1, pp. 503–528, 1989.

[72] S. Benson and J. More, "A limited memory variable metric method in subspaces and bound constrained optimization problems," *mathematical, Information and Computer Science Division, Argonne National Laboratory, ANL/MCS-P*, vol. 901, 2001.

[73] R. Byrd, P. Lu, J. Nocedal, and C. Zhu, "A limited memory algorithm for bound constrained optimization," *SIAM Journal on Scientific Computing*, vol. 16, no. 5, pp. 1190–1208, 1995.

[74] J. Bezdek and R. Hathaway, "Convergence of alternating optimization," *Neural, Parallel and Scientific Computations*, vol. 11, no. 4, pp. 351–368, 2003.

[75] M. Qian and C. Zhai, "Robust unsupervised feature selection," in *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pp. 1621–1627, AAAI Press, 2013.

[76] D. Luo, C. Ding, and H. Huang, "Towards structural sparsity: an explicit l2/l0 approach," in *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pp. 344–353, IEEE, 2010.

[77] Q. Gu, M. Danilevsky, Z. Li, and J. Han, "Locality preserving feature learning," in *International Conference on Artificial Intelligence and Statistics*, pp. 477–485, 2012.

[78] F. Nie, H. Wang, H. Huang, and C. Ding, "Adaptive loss minimization for semi-supervised elastic embedding," in *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pp. 1565–1571, AAAI Press, 2013.

[79] M. Qian, F. Nie, and C. Zhang, "Efficient multi-class unlabeled constrained semi-supervised svm," in *Proceedings of the 18th ACM conference on Information and knowledge management*, pp. 1665–1668, ACM, 2009.

[80] M. Qian, F. Nie, and C. Zhang, "Probabilistic labeled semi-supervised svm," in *Proceedings of the 2009 IEEE International Conference on Data Mining Workshops*, pp. 394–399, IEEE Computer Society, 2009.

[81] M. Qian, B. Chen, H. Xu, and H. Qi, "How about utilizing ordinal information from the distribution of unlabeled data," in *Proceedings of the 19th ACM international conference on Information and knowledge management*, pp. 289–298, ACM, 2010.

[82] M. Qian, "Text-image topic discovery for web news data," in *Advances in Information Retrieval*, pp. 675–680, Springer, 2014.

[83] M. Qian and C. Zhai, "Unsupervised feature selection for multi-view clustering on text-image web news data," in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pp. 1963–1966, ACM, 2014.

[84] R. T. Rockafellar, *Convex analysis*, vol. 28. Princeton university press, 1997.

[85] D. Seung and L. Lee, "Algorithms for non-negative matrix factorization," *Advances in neural information processing systems*, vol. 13, pp. 556–562, 2001.

[86] M. Porter, "An algorithm for suffix stripping," *Program: electronic library and information systems*, vol. 14, no. 3, pp. 130–137, 1993.

[87] G. Pass, R. Zabih, and J. Miller, "Comparing images using color coherence vectors," in *Proceedings of the fourth ACM international conference on Multimedia*, ACM, 1997.

[88] H. Tamura, S. Mori, and T. Yamawaki, "Textural features corresponding to visual perception," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 8, no. 6, 1978.

[89] T. Lee, "Image representation using 2d gabor wavelets," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 18, no. 10, pp. 959–971, 1996.

[90] Y. Ro, M. Kim, H. Kang, B. Manjunath, and J. Kim, "Mpeg-7 homogeneous texture descriptor," *ETRI journal*, vol. 23, no. 2, pp. 41–51, 2001.

[91] S. Bickel and T. Scheffer, "Multi-view clustering," in *Proceedings of the Fourth IEEE International Conference on Data Mining*, pp. 19–26, IEEE Computer Society, 2004.

[92] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge Univ Pr, 2004.

[93] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural computation*, vol. 15, no. 6, pp. 1373–1396, 2003.